

Christian Döbel

**Ein Smart Home Management-System
basierend auf adaptive Lernalgorithmen
des Verhaltensserwerbs (ENKOS)**

**Ein Smart Home Management-System
basierend auf
adaptive Lernalgorithmen des
Verhaltensserwerbs (ENKOS)**

Christian Döbel



Universitätsverlag Ilmenau
2016

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über <http://dnb.d-nb.de> abrufbar.

Diese Arbeit hat der Fakultät Informatik und Automatisierung der Technischen Universität Ilmenau als Dissertation vorgelegen.

Tag der Einreichung: 9. März 2015

1. Gutachter: Prof. Dr.-Ing. habil. Christoph Ament
(Technische Universität Ilmenau)

2. Gutachter: Prof. Dr.-Ing. habil. Horst Puta
(Technische Universität Ilmenau)

3. Gutachter: Prof. Dr.-Ing. Silvio Bachmann
(Fachhochschule Schmalkalden)

Tag der Verteidigung: 8. Dezember 2015

Technische Universität Ilmenau/Universitätsbibliothek

Universitätsverlag Ilmenau

Postfach 10 05 65

98684 Ilmenau

www.tu-ilmenau.de/universitaetsverlag

Herstellung und Auslieferung

Verlagshaus Monsenstein und Vannerdat OHG

Am Hawerkamp 31

48155 Münster

www.mv-verlag.de

ISBN 978-3-86360-142-3 (Druckausgabe)

URN urn:nbn:de:gbv:ilm1-2015000646

Titelfoto: photocase.com

Danksagung

Ich bedanke mich sehr bei meiner Frau Caroline und meinen Töchtern Pauline sowie Amalia, die viele Stunden auf mich verzichten mussten, damit ich diese in die vorliegende Arbeit investieren konnte. Trotzdem waren sie auch oft dabei, um beispielsweise die Lichtschranken mit den selbst gebauten Schaltungen in unsere neuen Türen zu installieren oder die Kabel im gesamten Haus zu verlegen, um die Versuche im Eigenheim durchführen zu können. Außerdem konnte ich in meiner Frau oft einen geduldigen Zuhörer und zusätzlichen Impulsgeber für Ideen finden.

Außerdem gilt mein Dank Prof. Dr.-Ing. Christoph Ament für die Betreuung meiner Arbeit. Als ich ihm die Idee dieses wissenschaftlichen Ansatzes vorstellte, war er sofort begeistert und motivierte mich, diese zu verfolgen.

Ich danke der Firma Aizo für die Bereitstellung von anonymisierten Nutzerdaten aus mehreren Haushalten, mit denen ich meine eigenen abgleichen und zum „Normalhaushalt“ vereinen konnte. Auch brachten Fachgespräche direkt in Wetzlar entscheidende Erkenntnisse über das alltägliche Verbraucherverhalten in Haushalten.

Für die zeitliche Entlastung danke ich meinen Eltern Stefanie und Dipl.-Ing. (FH) Frank Döbel, die einsprangen, damit ich mich am Wochenende auf Simulationen konzentrierte und vor allem meinen Großeltern Marlis und Dipl.-Ing. (FH) Werner Hesse, die mir stets ein gutes Beispiel dafür sind, dass die menschliche Neugier niemals erlöschen oder der Angst vor Neuem weichen darf, wenn man das Leben aktiv gestalten möchte. Diese Neugier ist der wesentliche Grund für die vorliegende Arbeit.

Der eine wartet,
dass die Zeit sich wandelt.
Der andere packt sie kräftig an –
und handelt.

Dante Alighieri, 14. Jahrhundert

Zusammenfassung

Mit der vorliegenden Arbeit ist es gelungen, ein adaptives Lernsystem „ENKOS“ („Energie- und Komfortmanagementsystem“) auf Basis von Learning Classifier Systems zu entwickeln, um menschliche Verhaltensweisen in Bezug auf die Benutzung elektrischer Geräte in einem Haushalt zu lernen. Diese konnten für die Steigerung des Wohnkomforts durch eine intelligente Vorhersage eingesetzt werden. Der Lernalgorithmus bezieht zum einen die Korrektur der Entscheidung von ENKOS durch den Nutzer und zum anderen die Korrekturzeit ein und realisiert damit adaptives „Smart Home“-System mit einem völlig neuen Ansatz.

Die Informationen, die durch die Benutzungsmuster von elektrischen Geräten durch die Nutzer erhoben werden können, werden außerdem zum Aufbau von logischen und elektrischen Modellen der Wohnung benutzt, womit eine signifikante Energiereduzierung erreicht werden konnte. Beide Ziele, die Komfortsteigerung sowie die Energieminimierung in einem standardisierten Haushalt, wurden in eine Zielfunktion integriert und damit quantifiziert. Für die Anpassung der Zielfunktion wurden auch verhaltenspsychologische Methoden der menschlichen Entscheidungsfindung herangezogen, um die objektive Steuerstrategie aller Geräte im simulierten Haushalt in die subjektive Empfindung von Menschen überführen zu können.

Danach wurde die Topologie von ENKOS, dem zentralen Lern- und Steuersystem der elektrischen Geräte, aus mehreren Möglichkeiten ausgewählt. Insbesondere für die Integration der einfachen physikalischen Modelle in ein komplexes Verhaltensmodell (abgebildet durch die LCS-Regeln) wurden mehrere grundsätzliche Fragestellungen gegeneinander abgewogen, um eine entsprechende Gesamtopologie zu generieren. Die Zielvariante wurde

„Kognitive Learning Classifier Systems“ (KLCS) genannt, da die LCS die Grundlage des Ansatzes bilden und um Methoden der kognitiven Verhaltensforschung erweitert wurden.

Nach dem Aufstellen der Systemstruktur wurde eine Standardsimulationsumgebung geschaffen, um verschiedene Parameter des geschaffenen KLCS objektiv bewerten zu können, um die Zielfunktion zu maximieren. Dabei kam heraus, dass zwei von drei untersuchten Methoden der Energieminimierung aussichtsreich sind, wobei bei der Parametrisierung der LCS-Komponenten lediglich zwei von sechs Parametern einen signifikanten Einfluss auf den Zielfunktionswert haben.

Es konnte gezeigt werden, dass das aufgestellte System ENKOS sowohl eine Steigerung des Komforts für die Nutzer ermöglicht sowie die signifikante Reduzierung des Energieverbrauchs in der simulierten Standardwohnung realisiert. Damit ist ein wirklich adaptives Smart Home-System im Sinne des Anwenders als Ansatz geschaffen und evaluiert worden.

Abstract

In the present work it has been able to develop an adaptive learning system "ENKOS" ("Energy and Comfort management system") basing on Learning Classifier Systems to adopt human behavior in relation to the use of electrical appliances in a household. This could be used to improve living comfort through an intelligent prediction. The learning algorithm refers to a correction of a decision made by ENKOS by the user and as well as the correction time, and so realizes an adaptive "smart home" system with the help of a completely new approach.

The information got by the usage patterns of electrical equipment by users are also used for the construction of logical and electrical models of the

apartment, which could be used for a significant energy reduction in the household. Both of these goals, the increase of living comfort and the energy minimization in a standardized household were integrated into an objective function and thus quantified. For accurate parametrization of the objective function, behavioral psychological methods of human decision-making have been used to convert the objective control strategy of all devices in the simulated household into the subjective perception of people.

Thereafter, the topology of ENKOS, the central learning and control system of the electrical devices were selected from several options. In particular, for the integration of simple physical models into a complex behavior model stored in the LCS several fundamental issues were discussed to generate a final overall topology. The target variant was named “Cognitive Learning Classifier Systems” (KLCS) derived from the basis of the approach, the LCS, extended by the methods to adopt human decision making.

After setting up the structure of the system, a standard simulation environment was developed to evaluate different parameters of the created KLCS objectively in terms of maximizing the objective function. This revealed, two out of three tested methods of energy minimization and only two of six parameters of the LCS components are having a significant impact on the target function value.

It was shown that the established system ENKOS could both, an increase in comfort for the user and a significant reduction of energy consumption in the simulated standard apartment. With ENKOS a real adaptive smart home system was created to help the user in daily living.

Inhaltsverzeichnis

Danksagung	V
Zusammenfassung	VII
Abstract	VIII
Inhaltsverzeichnis	XI
1 Einführung	1
1.1 Themenbeschreibung	1
1.2 Motivation	1
1.3 Stand der Technik (Smart Home)	5
1.3.1 Bereits am Markt vorhandene Systeme im Smart Home-Bereich	5
1.3.2 Forschungsansätze adaptiver Smart Home-Systeme	9
1.4 Gliederung der Arbeit	16
2 Modellierung lernender Systeme	18
2.1 Grundlagen der Modellbildung	18
2.2 Klassifizierung von Modelltypen	19
2.3 Lernende Systeme	24
2.3.1 Klassifizierung lernender Systeme	24
2.3.2 Grundsätzliche Topologien lernender Systeme	30
2.3.2.1 Überwachtes Lernen	30
2.3.2.2 Bestärkendes Lernen	36
2.3.2.3 Unüberwachtes Lernen	42
2.3.3 Künstliche Neuronale Netze als Methode zur Verhaltensadaptierung	44
2.3.4 Learning Classifier Systems	45
2.3.4.1 Das Covering	52
2.3.4.2 Der Reinforcementvorgang	57
2.3.4.3 Pittsburgh- vs. Michigantyp	60
2.3.4.4 Weiterentwicklungen und Modifikationen von LCS	61
2.3.5 Die Building Blocks-Hypothese	69
2.3.5.1 Einfluss der Selektion	77
2.3.5.2 Einfluss der Rekombination	80

2.3.5.3	Einfluss der Mutation.....	82
3	Grundlagen der menschlichen Entscheidungsfindung.....	85
3.1	Normative Theorien.....	86
3.2	Präskriptive Modelle	86
3.3	Deskriptive (empirische) Theorien.....	88
4	Definition der Zielfunktion für ENKOS	94
4.1	Anforderungen an ENKOS und Abgrenzung zu bisherigen Systemen.....	94
4.2	Grundstruktur der Zielfunktion	94
4.3	Definition der Teilzielfunktion f_{comfort} für die Komfortmaximierung	96
4.3.1	Methoden für Teilziel Komfortmaximierung.....	106
4.3.2	Das Modell eines Entscheidungsprozesses	107
4.4	Definition der Teilzielfunktion $f_{\text{energy}}(t)$ für die Energieminimierung.....	110
4.5	Aufstellen der Zielfunktion f_{ENKOS} aus den Teilzielen	113
5	Die Simulation des Standardhaushalts und die Entwicklung von ENKOS	116
5.1	Grundsätzliche Versuche im eigenen Haus	116
5.2	Allgemeine Topologie der Simulationsumgebung.....	128
5.3	Die Simulation eines Normjahres als Referenz.....	131
6	Modellbasierte Entwicklung von ENKOS	132
6.1	Allgemeine Topologie des Zielsystems und Einbettung in dessen Umwelt	132
6.2	Systemintegration zum „Kognitiven Learning Classifier System“ (KLCS)	139
6.2.1	Einschränkung des Definitionsbereichs	141
6.2.2	Agenten dienen als virtuelle Nutzer	142
6.2.3	Aufruf eines externen Lehrers.....	143
6.2.4	Einbindung der analytischen Komponenten zur Energieminimierung	146
6.2.5	Generalisierung als menschliche Abstraktion	146
6.3	Abbildung menschlichen Verhaltens mit Verhaltenserwerbsmethoden.....	147
6.3.1	Lösungsansatz Neuronale Netze	147
6.3.2	Modellierung mit Hilfe von Learning Classifier Systemen	154
6.3.2.1	Modellierung mit Hilfe von Learning Classifier Systemen.....	154
6.3.2.2	Die Generalisierung zur Wissensverdichtung (Exploitation)	166
6.3.2.3	Umsetzung der Evolutionären Algorithmen zur Exploration	172
6.4	Implementierung Energieverbrauchsminimierung.....	177

6.4.1	Modellbasierte Nichtanwesenheitserkennung	177
6.4.2	Substitution von Verbrauchern	183
6.4.3	Widerspruchsunterdrückung	190
6.4.4	Zusammenfassung der Methoden zur Energieminimierung	194
7	Einfluss verschiedener Modellparameter auf den simulierten Zielfunktionswert ..	196
7.1	Einfluss der Lernrate	196
7.2	Einfluss der Mutationsrate	198
7.3	Einfluss der Generalisierungsrate	201
7.4	Einfluss der Generalisierungsrate der Lichtstärke	203
7.5	Einfluss der Generalisierung des Uhrzeitbereichs	206
7.6	Einfluss der Vergessensrate	208
7.7	Bewertung der Einflüsse	211
8	Ausblick	212
	Anlage A – Definition Kommunikationsprotokoll	214
	Anlage B – Kommunikationsdiagramm ENKOS/ Treiberapplikation	215
	Anlage C – Neuronale Netze	216
	Literaturverzeichnis	232

1 Einführung

1.1 Themenbeschreibung

In der vorliegenden Arbeit wird ein Smart Home-System entworfen, entwickelt und mit Hilfe einer Simulation charakterisiert, das über den Stand der Technik hinaus das Verhalten der Verbraucher in einem Haushalt lernt, um adaptiv dessen Wünsche bezüglich der Steuerung der Geräte gerecht zu werden. Das wird durch eine zentrale Steuerung aller Geräte im Haushalt realisiert. Außerdem ist Ziel des Systems die Minimierung des Gesamtenergieverbrauchs aller Geräte.

Diese beiden Ziele, die als Komfortmaximierung und Energieminimierung bezeichnet werden, können in einigen Situationen im Alltag komplementär sein, beispielsweise wenn der Fernseher in einem leeren Raum durch das System abgeschaltet wird, oder die beiden Ziele können sich entgegenstehen. Letzteres ist zum Beispiel der Fall, wenn der Anwender unbedingt eine Lampe mit großem Energieverbrauch einschaltet, während das System eine energiesparende auswählen würde.

Deshalb wird im ersten Schritt eine Zielfunktion generiert, um einerseits beide Ziele vergleichbar zu machen und andererseits mit Hilfe eines Vergleichsmaßes den Einfluss der einzelnen Parameter auf die Zielfunktion messen zu können. Damit ist die Optimierung des Systems möglich.

1.2 Motivation

Sowohl die Senkung des Energieverbrauchs als auch die Steigerung des Komforts in modernen Haushalten sind zwei Kernziele heutiger Entwicklungen, die häufig unter Smart Home-Systeme zusammengefasst werden. Ein großer Wohnkomfort ist dann gegeben, wenn mit wenig manuellen Ein-

griffen elektrische Geräte im Haushalt im Sinne des Verbrauchers geschaltet werden, beispielsweise das Licht beim Einbruch der Dämmerung ohne Nutzereingriff eingeschaltet wird.

In einer Studie der Hochschule Rosenheim [107] wird ein durchschnittliches Einsparpotential von über 20 % an elektrischer Leistung in einem Einfamilienhaus (unter Zuhilfenahme der EN 15232 [95]) identifiziert, wenn bisher am Markt vorhandene und noch zu entwickelnde Smart Home-Komponenten eingesetzt werden [59]. Zum gleichen Ergebnis kommt auch Prof. Reiner Hirschberg in seiner Studie [111].

In der zitierten Studie selbst sind Vorschläge für Smart Home-Komponenten gemacht, wie beispielsweise das automatisierte Schalten der Beleuchtung in Wohngebäuden oder die Maximierung des Lichteinfalls in Gebäude durch eine intelligente Steuerung, um elektrisches Licht so selten wie möglich einzuschalten. Auf eine intelligente Steuerung sämtlicher Geräte in Abhängigkeit menschlicher Verhaltensweisen wird nicht direkt eingegangen, allerdings wird damit das Erreichen einer besseren Energieeffizienzklasse nach EN 15193 im Bereich Licht- und Hilfsenergie gezeigt.

Auf der Seite des Gesetzgebers definiert das „20 20 20“-Ziel der EU die europaweiten Einsparziele im Bereich Energieverbrauch von Gebäuden bis zum Jahr 2020:

- ➔ 20 % weniger Energieverbrauch von Gebäuden gegenüber 1990,
- ➔ 20 % weniger Treibhausgasausstoß gegenüber 1990 sowie
- ➔ 20 % Anteil der erneuerbaren Energien am Gesamtenergieverbrauch.

Um dieses Ziel zu erreichen, hat die EU mit der „Energy Performance of Buildings Directive“ [33] eine Richtlinie zur Verbesserung der Energieeffi-

zienz in Gebäuden erlassen, die die Mitgliedsstaaten bis 2006 in nationale Gesetze umzusetzen hatten. In dieser Richtlinie wurden nicht nur die Berechnungsvorschriften für die Gebäudeeffizienz an sich definiert, sondern auch Mindestanforderungen für neue Gebäude festgeschrieben. Desweiteren fordert die Richtlinie die in Deutschland bereits üblichen Energieausweise für Gebäude sowie die Überprüfung von älteren Heizanlagen. Eine weitere Richtlinie, die bis 2030 eine Reduzierung des Stromverbrauchs von 30 % gegenüber 1990 fordert, ist bereits in Vorbereitung, nachdem sich die 28 EU-Kommissare einvernehmlich darauf geeinigt haben [37].

Auch steigt die Nachfrage an energiesparenden Systemen im Haushalt permanent, da sich der Strompreis in Deutschland in den vergangenen 15 Jahren um durchschnittlich 4,5 % jährlich auf inzwischen über 0,28 €/kWh verteuert hat [7]. Eine Änderung dieses Trends ist gerade vor dem Hintergrund der Energiewende nicht zu erwarten.

So ergab eine Studie der Fachhochschule Köln und Jung Schaltersysteme eine sehr große Kaufbereitschaft von 80 % der Deutschen für Smart Home-Systeme, wenn die Ziele Energieeinsparung und Erhöhung der Sicherheit prinzipiell adressiert werden [14]. Größtes Hemmnis sind dabei die fehlenden Komplettlösungen inklusive der Montage- und Servicedienstleistungen auf dem Markt. Während bereits Systeme mit adaptiven Fähigkeiten erforscht wurden, die das Anwenderverhalten abbilden, gibt es noch keine konsequenten, ganzheitlichen Ansätze, bei denen die Arbeitspunkte sämtlicher Geräte zentral eingestellt werden.

Tatsächlich nutzen 2 bis 3 % der deutschen Verbraucher¹ eine Smart Home-Lösung [96], während 50 bis 60 % der Deutschen ein Interesse an einem solchen System haben [14]. Das zeigt den großen Bedarf am Markt, wenngleich für 38 % der Deutschen die Kosten am Markt erhältlicher Systeme noch zu groß sind und 28 % noch nicht an die nötige Reife der Technik glauben [14]. Gleichzeitig berichten zwischen 81 und 89 % der Verbraucher in Deutschland laut einer Bitkom-Umfrage einen Mangel an Informationen (im Sinne des Marketings) über Smart Home-Systeme [14].

Aufgrund der zunehmenden Breitbandversorgung in Europa prognostiziert Deloitte eine Verdopplung des Umsatzes mit Smart Home-Komponenten, von derzeit 1,7 Mrd. € bis 2017 [109]. Um diese Entwicklung gezielt zu fördern und Deutschland strategisch zu einem Leitmarkt im Smart Home-Bereich zu entwickeln, gibt es im Bundesministerium für Wirtschaft und Technologie seit 2011 einen Arbeitskreis zu diesem Thema. Ziel ist es, Deutschland zum Leitmarkt im Bereich Smart Home zu entwickeln. Als große Herausforderung auf dem Weg dorthin wird derzeit die fehlende Standardisierung gesehen, die von namhaften Vertretern in diesem Bereich, beispielsweise Dirk Wössner von der Deutschen Telekom, auch nicht in naher Zukunft gesehen wird.

Während der Smart Home-Ansatz auf den Einsatz im häuslichen Bereich zielt, wird im industriellen Bereich als Pendant dazu an der Smart Factory geforscht. Kern ist ebenfalls die ganzheitliche Optimierung, die zentrale Steuerung der Maschinen und Anlagen zur Energieeinsparung [73].

¹ Als Verbraucher werden nachfolgend die Menschen bezeichnet, die die Geräte im Haushalt nutzen. Elektrische Geräte erbringen einen Service für die Verbraucher, beispielsweise stellt das Radio akustische Informationen für Verbraucher zur Verfügung.

Eine Vision für Smart Home-Systeme für die Energieversorgung wurde im Rahmen der „Energiewende“ durch Prof. Peter Birkner vorgestellt [15]. Während in der ersten Phase bis 2020 die Spitzenleistung auf der Erzeugerseite, vor allem der Photovoltaikmodule und der Windkraftanlagen permanent ausgebaut wird, wird im nächsten Jahrzehnt aufgrund des technologischen Fortschritts sowie der fehlenden Wirtschaftlichkeit eines weiteren Ausbaus allein für Worst Case-Szenarios das Demand-Side-Management die größte Rolle spielen. Nicht nur für den Verbraucherkomfort, sondern vielmehr, um die Netzstabilität zu gewährleisten, werden die Optimierungsroutinen von den Stromnetzen auf die Verbraucher nach einer internen Mainova-Studie ausgeweitet werden müssen, die dann im Rahmen eines Smart Home-Ansatzes zentral gesteuert werden. Der Verbraucherkomfort ist dabei das Maß der korrekten Ansteuerung von elektrischen Geräten im Haushalt im Sinne des Verbrauchers.

1.3 Stand der Technik (Smart Home)

1.3.1 Bereits am Markt vorhandene Systeme im Smart Home-Bereich

Am Markt sind bereits mehrere Smart Home-Systeme vorhanden, weitere sind nur als Studie veröffentlicht worden. Aufgrund einer fehlenden Standarddefinition für Smart Home handelt es sich bisher in den meisten Fällen lediglich um Systeme, die elektrische Geräte fernsteuern können bzw. die das Auslesen von Stromverbrauchszählern durch den Energieversorger ermöglichen. In der vorliegenden Arbeit wird von einem Smart Home gesprochen, wenn das System, bestehend aus allen Geräten im Haushalt, adaptiv und zentral gesteuert wird, ohne dass ein Nutzereingriff nötig ist.

Eines der bekanntesten und viel umworbenen Systeme, das bereits am Markt erhältlich ist, ist das „Smarthome“ von der Firma RWE Effizienz GmbH, einer Tochterfirma des Energieversorgers RWE [106]. Dabei sind Jalousien zur Verschattung von Räumen, die Beleuchtung, die Raumtemperaturregelung sowie die Steckdosen fernschaltbar. Direkte Verbindungen zu Aktoren oder Sensoren wie beispielsweise eine Messung der Raumtemperatur sind dabei nicht möglich, da das System selbst als geschlossenes entwickelt wurde. Eine Konfiguration der Schaltmuster muss online durchgeführt werden, da kein Lernalgorithmus implementiert ist.

Ein weiteres, weit verbreitetes System ist „Joonior“ von der Firma Telefunken [123]. Auch hierbei wird die Raumautomatisierung durch eine Programmierschnittstelle durchgeführt, wobei allerdings die Bedienung im Wesentlichen per Smartphone-App erfolgt. Die technologische Grundlage bildet das Übertragungssystem von EnOcean, in das sich, im Gegensatz zum RWE Smarthome, Sensoren integrieren lassen. Damit sind komplexere Regelalgorithmen möglich, allerdings ist ebenfalls keine Lernkomponente vorgesehen.

Ebenfalls auf die EnOcean-Technologie basiert das „GFVS-Save“ von Eltako Electronics [32], wobei Eltako sich im Wesentlichen auf die Software und HMI-Schnittstelle konzentriert. Das System ist zur Steuerung durch einen PC vorgesehen, was es von den beiden vorher genannten unterscheidet. Im Gegensatz dazu ist „Home Matic“ von eQ-3/ ELV [34] ein geschlossenes Funksystem mit eigenem Protokoll, an das fast beliebig Sensoren und Aktoren angeschlossen werden können. Sowohl die Heizungs- als auch die Steuerung der elektrischen Geräte ist per Programmierung, vorzugsweise per App, möglich, neben dem Ver- und Entriegeln der Außentüren. Auch

Alarmfunktionen und Beleuchtungsszenarien sind programmierbar, eine aktive Lernkomponente ist nicht vorgesehen.

Ein weiteres System ist „DigitalStrom“ von AIZO [1], das sich wachsender Marktanteile erfreut. Hierbei ist die Kernkompetenz die Signalübertragung per Phasenanschnitt der im Haushalt üblichen 230 V-Wechselspannung (vgl. Abbildung 1). Dabei werden alle angeschlossenen elektrischen Verbraucher steuerbar und deren aktuelle Stromaufnahme auslesbar.

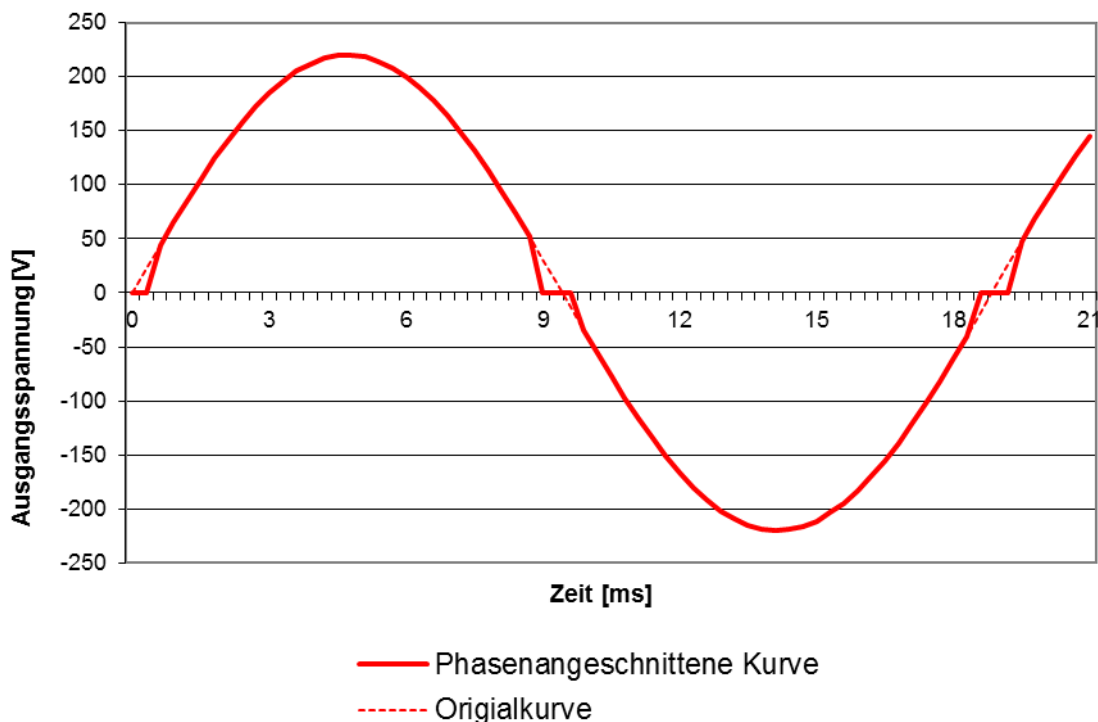


Abbildung 1. Phasenanschnitt der Spannung [eigene Darstellung].

Der wesentliche Unterschied zu den bisher dargestellten Systemen ist das dezentrale Konzept, bei dem ein beliebiger Lichtschalter oder eine Fernsteuerung in die gesamte Elektroinstallation leitungsgebunden die Nummer eines Wunschszenarios einspeist. Jeder Verbraucher hat für dieses Wunschszenario seinen bestimmten Zustand in eine intelligente Stromklemme einprogrammiert (s. Abbildung 2), den es daraufhin einnimmt. Da-

mit ist keine zentrale Instanz nötig. Ein optionaler Mini-Server sitzt auf der Hutschiene im Elektroanschlusskasten und bildet die Schnittstelle zum Hausstromnetz sowie zum Internet, damit alle aktuellen Zustände der Verbraucher auch im Internet sichtbar sowie damit steuerbar werden.



Abbildung 2. Intelligente Klemme der Firma Aizo (DigitalStrom) [3].

Der „Homeserver“ von Gira ist ebenfalls auf die Hutschiene montierbar und kommuniziert mit den elektrischen Verbrauchern per KNX/ EIB-Schnittstelle, um verschiedene Muster an Ansteuerungen einstellen zu können [40]. Ein Anschluss von Sensoren oder Aktoren ist nicht möglich. Sehr ähnlich ist das Konzept von „domovea“ von Hager [44] und vom „Miniserver“ der Firma Loxone [79]. Hier findet jeweils die Kommunikation per KNX-Bus statt, die Programmierung über einen Server, der auf die Hutschiene aufgesteckt wird. Entweder per Internet oder per App ist die Schnittstelle zum Anwender möglich.

Das System „TaHoma“ von Somfy ist ein recht vielfältiges System zur Raumautomation [113], da mit io-homecontrol ein eigenes Funkprotokoll benutzt wird. Dabei existiert eine Schnittstelle zu den Komponenten von EnOcean sowie zum Internet, über das die Schnittstelle zum Anwender gebildet wird. Damit werden die elektrischen Verbraucher sowie die Tempera-

tur programmiert, doch auch wie bei den vorangegangenen Systemen ist eine Programmierung nötig.

Mit dem Fokus auf die Verschattung von Gebäuden durch die Steuerung der Rollläden ist der „Homepilot“ von Rademacher entwickelt worden, wenngleich inzwischen auch andere Verbraucher per App oder Internet steuerbar sind [102]. Da das System ebenfalls mit einem eigenen Funkprotokoll arbeitet, ist keine Erweiterung mit Komponenten anderer Systeme möglich. Der Server wird an der Steckdose betrieben und mit dem Internet verbunden, während bei „MyGekko“ der Server in einem Touch-PC installiert wird und bei diesem Open-Source-System die Installation von Sensoren und Aktoren möglich ist [93]. Hier ist sogar die Raumtemperaturüberwachung durch Programmierung einer Zieltemperatur möglich.

Allen vorgestellten Systemen, die sich im Wesentlichen den Markt im Smart Home-Bereich derzeit teilen, ist gemeinsam, dass sie zum einen nicht erkennen, ob ein Anwender sich überhaupt im Raum befindet², und zum anderen keine eigene Intelligenz zur Erkennung der Nutzerwünsche besitzen, sondern aufwändig programmiert werden müssen.

1.3.2 Forschungsansätze adaptiver Smart Home-Systeme

Über käufliche Systeme hinaus sind im Bereich der adaptiven oder intelligenten Smart Home-Systeme bereits einige Forschungsprojekte durchgeführt worden, die nachfolgend dargestellt werden. Ein Beispiel dafür ist das

² Im Gegensatz zur optischen Raumüberwachung werden Systeme mit taktiler Sensorik (elektrisch, mechanisch) sowie kapazitiver Sensorik (elektrische Aufladung zur berührungslosen Erkennung) als auch Druck- sowie Radarsensoren im Rahmen von Forschungsprojekten eingesetzt. Desweiteren sind Ansätze für Temperaturmessungen möglich. Alle Systeme haben gemeinsam, dass eine absolut sichere Erkennung bislang nicht möglich ist.

adaptive Haus „ACHE“ [91], das in Boulder, Colorado, aufgebaut wurde. Die Motivation für dieses Projekt ist, dass die Akzeptanz von Smart Home-Systemen stark sinkt, wenn zur Inbetriebnahme eine umständliche Programmierung nötig ist, die mühsam an die sich verändernden Gewohnheiten der Anwender angepasst werden muss.

Hier wurde ein adaptives System entwickelt, das den Anwender eine gewisse Zeitdauer beobachtet, indem es dessen Steuerwünsche gegenüber Heizungsthermostaten, dem Warmwasserverbrauch und der Lampen in Abhängigkeit von dessen Anwesenheit in Räumen misst. Wenn das System ACHE sich sicher genug für eine eigene Vorhersage ist, steuert es diese Aggregate im Haushalt selbst. Der Nutzer kann die Vorhersage jederzeit korrigieren, indem er beispielsweise das Licht wieder ausschaltet, sollte es fälschlicherweise eingeschaltet worden sein. Die Anwesenheitserkennung in diesem Projekt wurde durch handelsübliche Bewegungsmelder realisiert.

Kern des Projekts ist es, die Kostenfunktion $J(t_0)$, die in (1) gezeigt ist, zu minimieren. Dabei fließt zum einen direkt der Term $d(x_t)$ ein, der die Unbequemlichkeit („discomfort“) in einem Zustand \mathbf{x} zum Zeitpunkt t beschreibt. Je unbequemer ACHE für den Nutzer ist (viele Fehlentscheidungen), desto größer wird $d(x_t)$. Die Zielfunktion besteht dabei aus κ kleinen Zeitbereichen.

$$J(t_0) = \mathbb{E} \left[\lim_{k \rightarrow \infty} \frac{1}{\kappa} \sum_{t=t_0}^{t_0+\kappa} d(x_t) + e(u_t) \right] \quad (1)$$

Zum anderen wird durch den Term $e(u_t)$ der Energieverbrauch bewertet, wenn eine Steuerentscheidung \mathbf{u} zum Zeitpunkt t getroffen wird. Daraus

ergibt sich als Optimierungsziel, eine Strategie zu entwerfen, die durchschnittlichen Gesamtkosten $J(t_0)$ in (1) zu minimieren.

Als Lernmethode zur Adaptierung des Verbrauchsverhaltens wird Q-Learning benutzt (basierend auf dynamischer Programmierung) [91]. Der Vorteil dieses Ansatzes des unüberwachten Lernens ist, dass die Lösung eine Gesamtstrategie ist, abhängig vom aktuellen Zustand, und dass der Ansatz immer gegen eine gültige Lösung konvergiert sowie dass das Lernen bereits mit wenigen Trainingsdatensätzen bei sofortiger Rückmeldung über den Erfolg der aktuell durchgeführten Aktion möglich ist. Jede Zustandsänderung wird dabei über einen Ereignistrigger erfasst und die Reaktion auf den sich einstellenden neuen Zustand über einen Q-Learning-Controller eingestellt. Am Beispiel der Beleuchtung ist die Systemtopologie in Abbildung 3 dargestellt.

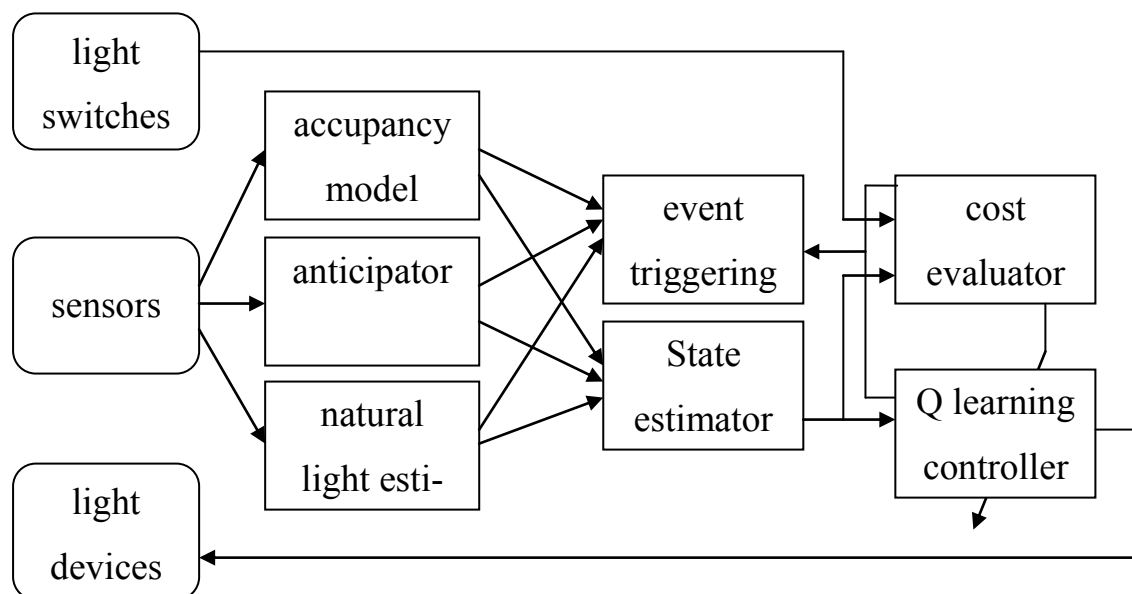


Abbildung 3. Topologie von ACHE am Beispiel der Beleuchtung [91]. Während auf der linken Seite die Lichtschaltersignale (light switches) sowie die Sensorinformation, aus denen ein Folgezustand antizipiert wird, eingelesen werden, wird per Q-Learning-Algorithmus das Optimum in der Kostenfunktion (s. (1)) ermittelt. Danach werden mit Hilfe des Arbeitspunkts die Lampen jeweils ein- bzw. ausgeschaltet.

Im Ergebnis wurde ACHE in einem Einfamilienhaus im Feldversuch getestet. Dabei konnte die Energieaufnahme aller Geräte reduziert werden, wobei die Unbequemlichkeit („discomfort“) auf hohem Niveau schwankte, wie in Abbildung 4 gezeigt ist.

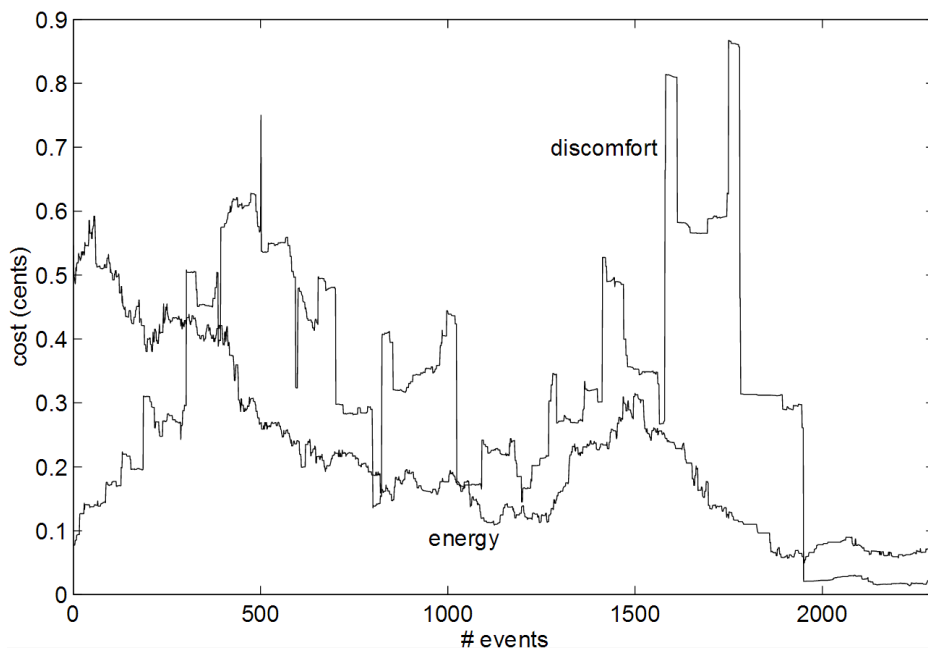


Abbildung 4. Ergebnis von ACHE [91]. Die Kostenfunktion $J(t_0)$ ist hier in seine Bestandteile $d(x_t)$ („discomfort“) sowie $e(u_t)$ („energy“) zerlegt. Ziel ist, diese zu minimieren, was für den Teil $e(u_t)$ gelingt.

Das bedeutet, dass eine fehlende Nutzerakzeptanz zu erwarten ist, da die Wünsche nicht ausreichend gut vorhergesagt wurden. Damit sollte der Q-Learning-Ansatz dahingehend weiterentwickelt werden, die Nutzerwünsche stärker zu berücksichtigen und beim Energiemanagement zu hinterfragen, ob ein ausschließlich adaptiver Ansatz ausreicht. Die Ursache ist darin zu sehen, dass Q-Learning das menschliche Verhalten nur bis zu einer bestimmten Abstraktionsebene abbilden kann. Diese ist nicht ausreichend, um die Komplexität menschlicher Handlungen zu adaptieren.

Ein weiterer Ansatz ist die Implementierung eines Neuronalen Netzes, um einen Smart Home-Haushalt aufzubauen [5]. Im untersuchten Ansatz sind zwei Neuronale Netze jeweils für die Energiereduzierung sowie die intelligente Verhaltensvorhersage eingesetzt: ein rekurrentes und ein Feed-Forward-Netz. Das rekurrente Netz wird zur Mustererkennung menschlicher Verhaltensweise im Haushalt eingesetzt, da die Stärken dieser Methode in der stark nichtlinearen Abbildung sequentieller und zyklischer Prozesse liegen. Dabei wird ein gesamter Tag als Zeitbasis benutzt und innerhalb dieser die Abfolge der Aktivitäten einer Person als einzelne aufeinanderfolgende Einträge gespeichert, als Lehrer eines überwachten Lernverfahrens. Das bedeutet, dass dem Neuronalen Netz die gesammelten Eingangs- und Ausgangsparameter am Ende des Tages präsentiert werden. Als Ergebnis soll dieses Netz menschliche Verhaltensweisen zu jedem Zeitpunkt anhand des aktuellen Kontextes voraussagen. Das Gütemaß ist dabei die Fehlerquadratsumme aus Vorhersage und vom Anwender eingestellten Zustand.

Das zweite Neuronale Netz (Feed-Forward-Architektur) soll die elektrische Leistungsaufnahme aller Geräte permanent minimieren. Hier geht es darum, eine möglichst effiziente Ansteuerung der elektrischen Verbraucher zu erreichen sowie ein Sicherheitssystem umzusetzen. Beide Netze werden mit denselben Eingangsparametern beaufschlagt, die Berechnung einer Systemantwort aus den Antworten beider Systeme ist im Beitrag [5] nicht gezeigt.

Die Simulation zeigt ein nicht zufriedenstellendes Ergebnis, da die Klassifizierung des komplexen menschlichen Verhaltens durch Neuronale Netze niedriger Ordnung³ nicht gelingt. Eine einfache Belehrung des Netzes, und zwar das Ein- oder Ausschalten des Ventilators in Abhängigkeit von der

³ Die Ordnung entspricht hier die Anzahl der Hiddenschichten.

Temperatur und der Luftfeuchtigkeit im Zimmer ist mit einem Netz ohne Hiddenschicht, also der Ordnung 0, realisiert. Da hier nur eine lineare Trennung möglich ist, kann diese Aufgabe nicht geleistet werden, wie aus Abbildung 5 hervorgeht. Dort ist mit „+“ der Ein- und mit „-“ der ausgeschaltete Ventilator dargestellt. Eine Trenngerade kann hier augenscheinlich kein Klassifikator sein.

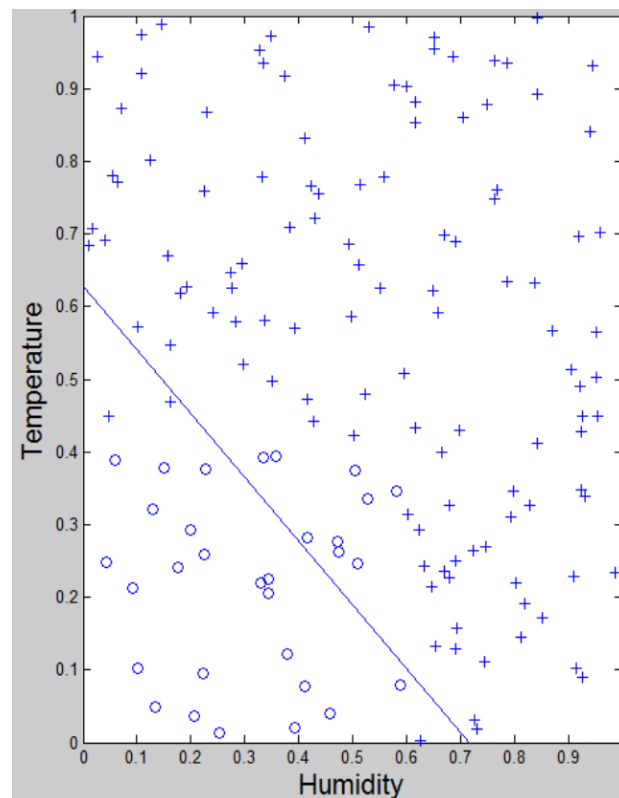


Abbildung 5. Simulationsergebnis Schaltung des Ventilators abhängig von Temperatur und Luftfeuchtigkeit [5]. Augenscheinlich ist ein Neuronales Netz 0. Ordnung nicht für diese Anwendung geeignet.

Da sich in der Simulation rasch Oszillationen einstellen, wurde auf eine Erhöhung der Ordnung verzichtet, womit die Verhaltensweise der Anwender nicht gelernt werden konnte. Im Beitrag wurde interessanterweise allerdings darauf hingewiesen, dass die Einbeziehung menschlichen Verhaltens im

Wesentlichen noch eine Lücke in der Forschungslandschaft im Smart Home-Bereich darstellt.

Ein weiterer Ansatz ist ein fuzzybasierter. Da die menschliche Entscheidungsfindung an sich regelbasiert abläuft (s. Kapitel 3), liegt es nahe, ein adaptives System aufzubauen, dass unscharfe Informationen erfasst, diese verarbeitet und eine Lösung als Kompromiss aus einzelnen, einfachen Regeln bildet [128]. Basis des Systems ist eine Lerntabelle, die online belehrt wird und deren Regeln sofort zum Einsatz kommen können. Im Rahmen der Fuzzyfizierung werden die drei Eingangsvariablen „Lichtstärke außerhalb“, „Anwesenheit“ und „Zeit“ in entsprechende Funktionen eingeordnet. Am Beispiel der Anwesenheit werden zwei Funktionen, nämlich abwesend („Absent“) und anwesend, („In Place“) implementiert, die in Abbildung 6 dargestellt sind. Dabei wird der gemessenen Anwesenheit jeweils ein Funktionswert für beide Funktionen in Abhängigkeit von der tatsächlichen Anwesenheit zugeordnet. Dieses im Beitrag gezeigte Beispiel ist jedoch eine Fuzzyfizierung eines diskreten Zustands, was in dieser Form nicht unbedingt sinnvoll ist.

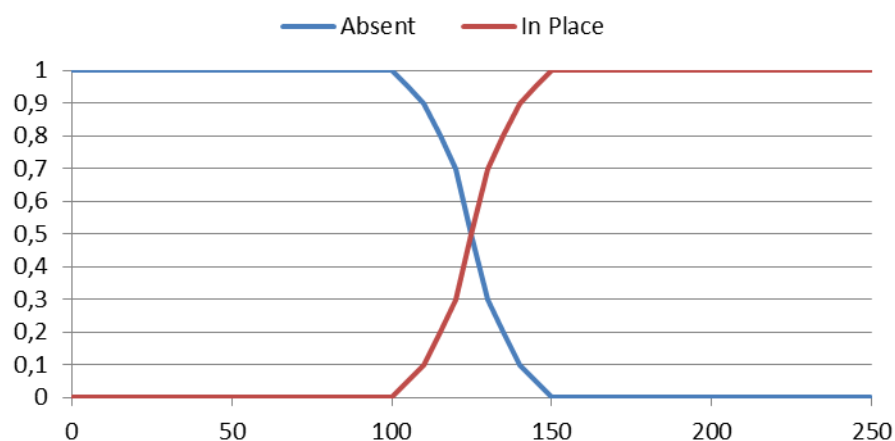


Abbildung 6. Fuzzyfizierung des Zustands Anwesenheit [128].

Nachdem die unscharfen („fuzzy“) Umweltinformationen in scharfe Funktionen verwandelt wurden, erfolgt die Berechnung des Ausgabezustands mittels Fuzzycontroller.

Das System befindet sich zuerst im eventbasierten Zustand, bevor der autonome Zustand erreicht wird, in dem das System komplett die Kontrolle über beide Aktuatoren übernimmt. In diesem Zustand kontrolliert der Anwender die Geräte direkt und das Fuzzysystem beobachtet diesen dabei. Auch hier gelingt es nicht, die Verhaltensadaptionen in einen akzeptablen Bereich zu bekommen. Hintergrund ist ebenfalls, dass mehrschichtige, nicht linear verknüpfte Regeln den Menschen als Entscheidungsgrundlagen dienen.

Zusammenfassend sind zwar Ansätze im Bereich adaptiver Smart Home-Systeme vorhanden, eine ganzheitliche Zustandsoptimierung aller Geräte in einem Haushalt mit hinreichender Nutzerakzeptanz ist jedoch in den vorgestellten Arbeiten nicht möglich gewesen. Diese Ansätze stützen sich dabei ausschließlich jeweils auf die gleiche Lernmethode, sowohl für die Komfortmaximierung als auch für die Energieminimierung.

1.4 Gliederung der Arbeit

Im zweiten Kapitel werden grundsätzlich die Möglichkeiten von Modellierungen als Voraussetzung für eine Optimierung dargestellt. Der Schwerpunkt liegt dabei auf den Modellen, die menschliches Verhalten adaptieren können, bevor die menschliche Entscheidungsfindung aus psychologischer Sicht im dritten Kapitel dargestellt wird.

Anschließend wird im vierten Kapitel die Zielfunktion, nach der das Modell für ENKOS aufgebaut wird, definiert. Nach der prinzipiellen Quantifizierung der Ziele Komfortmaximierung und Energieminimierung im vorlie-

genden Problem werden beide Terme entsprechend gewichtet, um daraus die Systemtopologie grundsätzlich zu entwickeln.

Im fünften Kapitel werden eine Simulationsumgebung für das zu entwickelnde System und damit ein Normalhaushalt zur Vergleichbarkeit der verschiedenen Ansätze geschaffen. Desweiteren wird die Schnittstelle zwischen der funktionalen Simulation als Umwelt und dem Zielsystem festgelegt, bevor im sechsten Kapitel das Modell ausgehend von der Zielfunktion sowie der groben Topologie gebildet wird. Dabei kommen sowohl Methoden des Verhaltenserwerbs als auch analytische Methoden und verschiedene Modelle zur Integration zum Einsatz. Dabei werden unter anderem Neuronale Netze mit den Learning Classifier Systems als Systemkern verglichen.

Nachdem die Struktur des Systems festgelegt und in die simulierte Umwelt eingebunden wurde, wird im Kapitel sieben der Zielfunktionswert in Abhängigkeit wesentlicher Modellparameter mit Hilfe eines Versuchsplans untersucht. Die Ergebnisse werden im Zusammenhang mit dem Modellansatz diskutiert. Abschließend folgt im achten Kapitel eine Zusammenfassung, bevor mögliche weitere Forschungsrichtungen und Anwendungen im Ausblick in Kapitel neun kurz erläutert werden.

2 Modellierung lernender Systeme

2.1 Grundlagen der Modellbildung

Mit den Methoden der Modellbildung soll einerseits das menschliche Verhalten abgebildet werden, um eine möglichst korrekte Vorhersage von Entscheidungen im Alltag durchführen zu können, andererseits der Energiehaushalt minimiert werden. Die grundsätzliche Struktur der Umsätze eines beliebigen Modells ist in Abbildung 7 gezeigt. Dabei fließen sowohl Energie (mechanische, elektrische, etc.), Signale (Sensorik) als auch Stoffe (Luft, Wasser, etc.) in das System hinein und auch wieder heraus. Im System selbst finden dabei Energiewandlung (z. B. mittels Energy Harvesting), Signalverarbeitung, wie etwa eine Regelung, sowie chemische, stoffverändernde Prozesse statt. Desweiteren sorgt eine Kommunikation zum Austausch von Informationen mit anderen Systemen über Bussysteme.

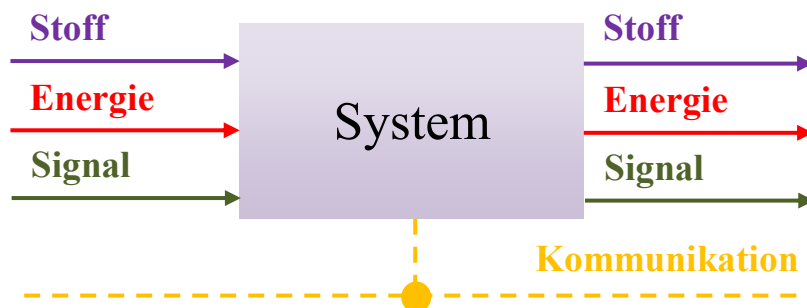


Abbildung 7. Grundstruktur eines Systems bezüglich seiner Umsätze [eigene Darstellung].

Auf der einen Seite müssen nicht alle Schnittstellen zwingend vorhanden sein, auf der anderen müssen nicht alle vorhandenen wirklich beschrieben werden. Wenn beispielsweise die Sichtweise auf die Umwelt bestimmte Betrachtungen nicht erfordert, werden diese im Modell zugunsten der Übersichtlichkeit auch nicht abgebildet. Der Extremfall, wenn gar keine Schnitt-

stellen vorhanden sind, spricht man vom geschlossenen, ansonsten vom offenen System [48] [57]. Eine Herausforderung ist dabei häufig, das System als Teil der Umwelt von vornherein klar von dessen Umwelt abzutrennen.

Außerdem wird bei der Modellierung zwischen dynamischen und statischen Systemen unterschieden. Bei dynamischen Systemen beeinflussen sich interne Zustände gegenseitig und können zu unterschiedlichen Ausgangsgrößen führen, auch wenn sich die Eingangsgrößen nicht ändern. Bei statischen ist das nicht möglich. Auch werden im Rahmen einer Modellierung diejenigen Systembestandteile intensiv beleuchtet, die wirklich für die gegebene Aufgabenstellung wichtig sind. Andere können weniger tief beleuchtet oder wiederum weggelassen werden, wie oben die geschlossenen Systemschnittstellen zur Umwelt [9].

In der vorliegenden Arbeit soll zum einen der elektrische Energiefluss in der Wohnung minimiert, zum anderen der Nutzerkomfort maximiert werden. Deshalb wird für die erste Teilaufgabe ein Energiemodell aller elektrischen Verbraucher in der Wohnung gebildet, während für die zweite Teilaufgabe ein logisches Modell (Verknüpfung aller elektrischen Geräte untereinander und in Bezug auf die Räume) sowie ein (psychologisches) Verhaltensmodell von den Nutzern (bezüglich deren Entscheidungsfindung, unter welchen Bedingungen Geräte eingeschaltet werden) gebildet wird.

2.2 Klassifizierung von Modelltypen

Zur Erstellung des Modells ist es wichtig, als erstes zu klären, was genau modelliert werden soll, da in der Regel ganz verschiedene Gesichtspunkte für die Entscheidung möglich sind. Deshalb ergibt sich der richtige Modell-

typ zum einen durch die Modellierbarkeit⁴ und zum anderen durch die konkrete Aufgabenstellung vorgegeben.

Allgemeine Eigenschaften	Darstellungsart	Verwendungszweck
Reale/ Ideale	Ikonische	Beschreibend
Natürl./ künstl.	Analoge	Simulationsmodell
Geschl./ offen	Symbolische	Erklärungsmodell
Statisch/ dynam.	Sprachliche	
Mechanische	Gedankliche	
	Gegenständliche	

Abbildung 8. Klassifizierung von Modelltypen [65].

In Abbildung 8 findet sich eine Zusammenstellung üblicher Modelltypen als eine Möglichkeit, ein geeignetes daraus auszuwählen. In der vorliegenden Arbeit lassen sich alle drei Teilmodelle, das Energiemodell, das logische Modell sowie das Verhaltensmodell, in die verschiedenen Sichtweisen einordnen. Die drei Modelle sind geschlossen, real und natürlich. Die Darstellung erfolgt symbolisch, wobei alle drei Modelle für die Simulation verwendet werden. Während das Energiemodell sowie das logische Modell statisch (unter der Annahme, dass keine Speicherelemente vorhanden sind) sind, handelt es sich beim Verhaltensmodell aufgrund der internen Zustände des menschlichen Denkens um ein dynamisches Modell.

⁴ Während beispielsweise einerseits einfache physikalische Zusammenhänge gut bekannt und aus der Literatur zu entnehmen sind, ist das für chemische Zusammenhänge nicht immer der Fall.

Nach der Auswahl des Typs wird die Sichtweise auf das System ausgewählt. In Abbildung 9 sind drei verschiedene Sichten beispielhaft gegenübergestellt. Dabei ist wiederum wichtig, was das Ziel der Modellbildung ist.

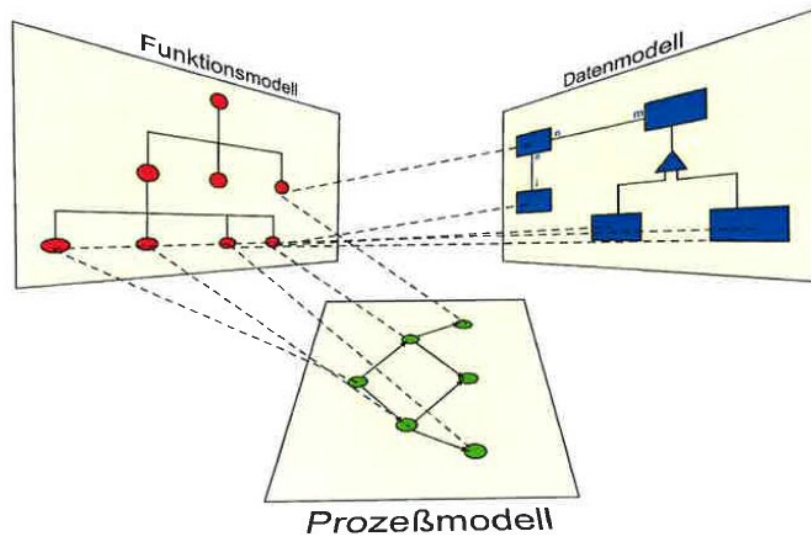


Abbildung 9. Verschiedene Sichtweisen auf ein Modell [65].

Zwei unterschiedliche Sichtweisen auf denselben Gegenstand sind in Abbildung 10 gezeigt. Auf der einen Seite würde die Optimierung des Verkehrsflusses oder des Lärms in der Straße die Fahrzeuge und Fußgänger in den Fokus beispielsweise einer Verkehrssimulation rücken, wie auf der rechten Seite der Abbildung gezeigt. Beim Smart Home-System würde das bedeuten, dass sich manche Teile des Verhaltensmodells beispielsweise auf die Beleuchtung in einem Haushalt konzentrieren, während andere auf die Benutzung von akustischen Informationsquellen zielen.



Abbildung 10. Zwei unterschiedliche Sichtweisen desselben Sachverhalts [146].

Wenn die grundlegende Modellierungsmethode und die Sichtweise passend zum Sachverhalt festgelegt sind, wird das Modell anhand von sechs Kriterien gebildet, die in [146] beschrieben sind. Für die vorliegende Problemstellung sollen später alle Kriterien im Rahmen der Modellierung der Systemstruktur erfüllt sein. Erst dann ist die Modellbildung abgeschlossen.

Als erstes soll die **Richtigkeit** durch das Modell gewahrt bleiben. Das bedeutet, dass das Modell die Verhaltensweisen korrekt im Sinne seines Aufbaus wie auch des Verhaltens abbildet (semantische Richtigkeit). Außerdem sollen nur für die Aufgabenstellung **relevante physikalische Größen** abgebildet werden. Wenn beispielsweise der Energiefluss eines Systems optimiert werden soll, sind die energetischen Ein- und Ausgangsgrößen nötig zu betrachten. Dazu wäre ein Modellansatz, der etwa Materialflüsse als Ein- und Ausgänge besitzt, ungeeignet (vgl. Modelltyp).

Desweiteren ist die **Wirtschaftlichkeit** der Modellbildung wichtig. Das soll verhindern, dass das Rad neu erfunden wird, sondern möglichst bekannte Modellierungstechniken als Voraussetzung benutzt werden. So gibt es in jeder der in Abbildung 8 gezeigten Modellierungsklassen bereits Ansätze, die übernommen werden können. Dieses Prinzip der Wirtschaftlichkeit sollte gewährleisten, dass bei gleichwertigen Modellansätzen stets der einfachere benutzt wird (Einfachheit).

Als weiteres Kriterium ist die **Klarheit** zu beachten. Dabei ist wichtig, dass das Modell auch vom Adressaten verstanden wird. Es sollen also interne Variablen und Zustände möglichst nach außen sichtbar und interpretierbar sein. In der Praxis kommt es daher hin und wieder an den Grenzen des Arbeitsbereichs vom Modell zu nicht erklärbaren Fehlentscheidungen. Das ist nicht selten darauf zurück zu führen, dass der Modellansatz nur eingeschränkt gültig ist, die Klarheit an den Systemgrenzen also nicht hinreichend gegeben ist [133]. Während in manchen Fällen Black Box-Ansätze unumgänglich sind, ist in anderen jedoch eine Alternative möglich. Am Beispiel eines Neuronalen Netzes lässt sich die schlechte Interpretierbarkeit der Wissensbasis zeigen. Eine mögliche Abhilfe ist die Integration eines Interpretationsumgebung für die Wissensbasis, die permanent das Netz in Vorwärtsrichtung mit verschiedenen Eingangsgrößen beaufschlagt und die Ausgangsgrößen entsprechend liest (als Ansatz des Reverse Engineering).

Im Sinne der **Vergleichbarkeit** sollen einheitliche äußere Schnittstellen definiert werden, damit verschiedene Modellierungsansätze miteinander in einem einheitlichen Benchmarkprozess miteinander verglichen werden können. Das bedeutet, dass verschiedene Modellansätze in die gleiche Lernplattform eingebettet und somit deren Vor- und Nachteile gezeigt werden können. Gerade das Verhalten in den Grenzbereichen ist wichtig zu kennen, als Vergleichsgröße wird in der Regel der aufintegrierte Wert der Zielfunktion nach einem bestimmten Zeitintervall genutzt. Dieser Punkt wird in der Praxis hin und wieder unterschätzt, womit die nachträgliche Änderung von Schnittstellen schwer möglich ist.

Und als letztes Kriterium soll das Modell einen **systematischen Aufbau** besitzen. Hintergrund ist, dass ein Modell einen bestimmten Teil der Realwelt widerspiegelt, bei einer Erweiterung (Modellbildung eines größeren

Teils der Realwelt) oder Einkürzung jedoch die Struktur selbst nicht verloren gehen darf. Desweiteren gewährt ein systematischer Aufbau die modulare Zusammensetzung von Teilmodellen, wie zum Beispiel analytisch beschreibbaren Abbildungen mit Neuronalen Netzen, zu einem Gesamtmodell.

2.3 Lernende Systeme

2.3.1 Klassifizierung lernender Systeme

Das Lernen ist nicht eindeutig in der Literatur definiert. Eine sehr bekannte Definition ist die von Kunz, wo Lernen eine „[...] Erfahrung basierende, relativ stabile Veränderung der Wahrscheinlichkeit zur Durchführung eines speziellen Verhaltens“ [69] ist. Eine andere bekannte (nach Simon) besagt: „Lernen ist jede Veränderung eines Systems, die es ihm erlaubt, eine Aufgabe bei der Wiederholung derselben Aufgabe oder einer Aufgabe derselben Art besser zu lösen“ [89]. In der Aufgabenstellung dieser Arbeit werden lernende Algorithmen eine zentrale Rolle spielen, um menschliche Verhaltensweisen in einem Smart Home-System adaptieren zu können. Eine Definition des Lernens auf Basis der Modellbildung ist in Definition 1 zitiert und bildet in der vorliegenden Arbeit die Grundlage für die Weiterarbeit.

„**Lernen** ist ein Prozess, bei dem ein System eine abrufbare Präsentation von vergangenen Interaktionen mit seiner Umwelt aufbaut“ [89].

Definition 1. Lernen.

Dabei ist zu beachten, dass gelernte Reaktionen erst bei Wiederauftreten der passenden Situationen abgerufen werden, womit die Lücke zwischen unbekannten Repräsentationen vernachlässigt wird. Das ist bei einer hinreichend

guten Wissensbasis der Fall. Davon ausgehend wird das **maschinelle Lernen** in Definition 2 definiert.

Das **maschinelle Lernen** ist „ein sich selbst belehrender Algorithmus, durch den aus einem Set von Eingangsgrößen (modellbasiert) ein Ausgangsgrößenvektor gebildet wird“ [4].

Definition 2. Maschinelles Lernen.

Im Gegensatz dazu existieren bereits zahlreiche einfache, nicht lernende Algorithmen, um definierte Aufgaben rechnergestützt durchzuführen, wie beispielsweise das Sortieren von Zahlen. Beim maschinellen Lernen sind der Algorithmus oder die zu verwendenden Parameter am Anfang der Lernaufgabe nicht exakt bekannt, es gibt jedoch zahlreiche Beispiele beim überwachten Lernen, die zum Aufbau einer internen Struktur (eines Klassifikators) genutzt werden können, oder die Umwelt bewertet das aktuelle lernende System beim bestärkenden Lernen. Eine mögliche maschinelle Lernaufgabe ist dabei ein Spamfilter, der sich permanent an zu blockierende Muster in E-Mails anpassen muss.

Während bei der Behavioristischen Lerntheorie das Lernende System als Black Box betrachtet wird, stehen bei Kognitiven Lerntheorien die Mechanismen des Modells selbst im Vordergrund und werden erforscht [108]. Bei den Sozialen Lerntheorien werden beide vorher genannten Ansätze zusammengefasst und die Wechselwirkung mit der Umwelt betrachtet.

Bei der Auswahl des optimalen Lernalgorithmus' müssen mehrere Faktoren, die aus der Problemstellung her rühren, beachtet werden. Als erstes ist es wichtig, wie das Modell in die vorhandene Umwelt integriert werden kann. Dabei sind entweder Trainingsdaten vorhanden, also Eingangs- und die da-

zugehörigen Ausgangsdaten, mit Hilfe deren das Modell gebildet wird (**überwachtes Lernen**). Oder alternativ dazu muss das Modell online gebildet und belehrt werden, sodass die Umwelt nur auf aktuelle Reaktionen des Modells eine Antwort formulieren kann (**bestärkendes Lernen**), oder es liegt überhaupt keine Rückkopplung zur Umwelt vor (**unüberwachtes Lernen**), sodass die Antwort auf anderem Wege ermittelt werden muss. Da diese Ansätze wesentlich sind, um die Verhaltensweisen menschlicher Verbraucher in der Wohnung zu lernen, wird im Abschnitt 2.3.2 genauer darauf eingegangen.

Nachdem die Methode des Lernansatzes bestimmt wurde, ist die nötige Komplexität des Modells zu ermitteln. Dabei ist wichtig, dass die Ordnung des Modells etwa diejenige der Umwelt widerspiegelt. Wenn die Umwelt beispielsweise sehr komplex ist, das Modell jedoch zu einfach, kommt es schnell zum Underfitting. Das bedeutet, dass das Modell nur einen kleinen Bereich, je nach Startpunkt, der Umwelt abbilden kann. Im entgegen gesetzten Fall spricht man vom Overfitting, wenn das Modell deutlich komplexer angesetzt wird als die Umwelt. Dann werden Störungen und Rauschen mitgelernt und als Signalbestandteile interpretiert [4].

Wie in Abbildung 11 dargestellt ist, gibt es im Allgemeinen verschiedene Abstraktionsebenen von Modellen. Bei der Modellierung sollte die Abstraktion so gering wie möglich gehalten werden, um die Umwelt möglichst genau abzubilden. Während ein mentales Modell nur sehr abstrakt, beispielsweise durch Sprache, die Umwelt beschreibt, etwa in klinischen Anwendungsfällen, stellt ein analytisches Modell die genaueste Form der Modellierung dar. Hier werden exakte physikalische Zusammenhänge durch mathematische Formeln eindeutig ausgedrückt.

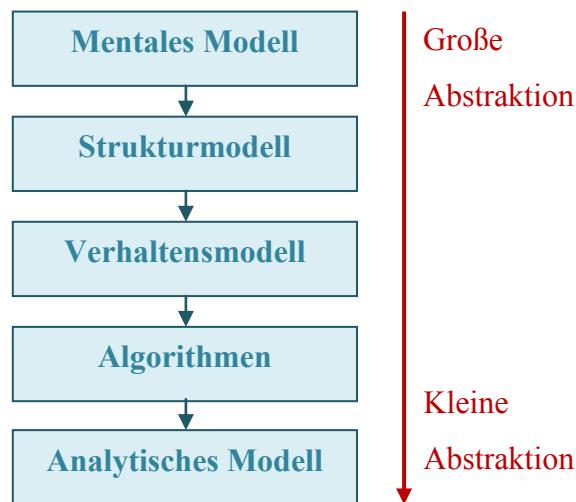


Abbildung 11. Abstraktionsebenen bei der Modellbildung (nach [115]).

Um die richtige Abstraktionsebene des Modells zu finden, wird die Methode der Abnehmenden Abstraktion [47] benutzt. Von der maximalen Abstraktion ausgehend muss der Punkt gefunden werden, an dem bereits genügend Realitätsnähe zum vorhandenen Problem vorhanden ist („relevance“ nach [94]), aber auch noch ein möglichst großer Anwendungsbereich erfasst wird (wissenschaftliche Strenge „rigour“) [77].

Des Weiteren ist sehr wichtig für die Modellbildung, wie viele Lerndaten zur Verfügung stehen. Kann man sich beispielsweise in kurzer Zeit mit wenig Aufwand (Wirtschaftlichkeit) viele Trainingsdaten durch die Beaufschlagung der Umwelt und Rücklesen der Antwort generieren, kommen andere Möglichkeiten in Betracht als bei wenigen Lerndaten, die nur teuer und langwierig zu erhalten sind (wenn z. B. erst komplexe Versuchsmuster gebaut werden müssen). Damit verbunden ergibt sich das Problem, dass die vorhandenen Lerndaten den gesamten Situationszustandsraum hinreichend weit repräsentieren müssen, damit spätere Daten in der Anwendungsphase nicht völlig neue Arbeitspunkte repräsentieren. In dieser Arbeit stehen eini-

ge Hundert Datensätze von Testtagen zur Verfügung, die teilweise im Eigenheim ermittelt wurden und zum anderen Teil von Fa. Aizo zugeliefert wurden [2].

Diese Randbedingungen, die für die Modellbildung geklärt werden müssen, ergeben ein Spannungsfeld für die Modellbildung, das in [26] erklärt ist. Zusammenfassend bestimmen dabei diese drei Faktoren die Wahl des Lernalgorithmus:

- Die **Komplexität der Hypothese** (z. B. die größte Ordnung des gewählten Polynoms), also der Umwelt, die modelliert werden soll;
- Die nötige **Anzahl an Trainingsdaten** und
- Der **Generalisierungsfehler** für neue Datensätze (also die Robustheit des Ansatzes).

Dabei stehen diese drei Ziele im Wettbewerb zueinander, da beispielsweise die Kapazität (also die maximal mögliche Komplexität) des Lernverfahrens so nah an der Umwelt wie möglich liegen sollte, was meist sehr viele Lerndaten benötigt, die ökonomisch nicht immer einfach zu beschaffen sind. Dabei sollen aber auch nicht alle verfügbaren Informationen zum Lernen benutzt werden, damit zusammen eine Validierungs- wie auch eine Testphase des Systems möglich ist.

Der Hintergrund ist, dass meist etwa die Hälfte der vorhandenen Datensätze (nach zufälliger Wahl) zum Belehren genutzt werden und circa ein Viertel zur Validierung des gefundenen Modellansatzes. Sobald das beste Modell mit der größten Generalisierungsfähigkeit gefunden wurde, wobei auch mehrere Rekursionen von Belehrung und Validierung in Form der Kreuzvalidierung (also stochastisches Neusortieren der Lern- und der Validierungsdatensätze) durchgeführt werden können, wird die Güte des Mo-

dells an sich geprüft. Dabei wird das Modell mit dem verbleibenden Viertel aller bislang nicht benutzten Datensätzen, beaufschlagt und der Gesamtfehler als (quadratisch) normierte Summe aller absoluten Abweichungen zwischen Modellausgang und Sollgröße berechnet.

Desweiteren ist es wichtig, bei der Entwicklung lernender Systeme den Grad der Autonomie zu definieren. Eine gute Einteilung von adaptiven bis hin zu sich selbst organisierenden Systemen bietet die Firma IBM [116]. Dabei ist das System genau dann gegenüber einem definierten Eingangsvektor **adaptiv**, wenn die Beziehung in (2) gilt, wobei „W die Menge der zeitabhängigen Funktionen ist, die ein Akzeptanzkriterium angibt, welches die Leistungsfunktion einhalten muss. Dabei entsteht bereits eine robuste Systemstruktur, bei der ein menschliches Eingreifen nur in Sonderfällen nötig ist“ [136], während bei **prädiktiven** Systemen nach der Einteilung durch IBM als nächst untere Ebene der Autonomie von Systemen der menschliche Bediener selbst noch die Entscheidung trifft, das System aber Handlungsempfehlungen gibt [58].

$$\forall(i \in I): \quad o = \beta(i) \quad \Rightarrow \quad p(i, o) \in W \quad (2)$$

Zu den adaptiven Systemen gehört auch das angestrebte Smart Home-System in dieser Arbeit, das in Definition 3 beschrieben ist.

In der vorliegenden Arbeit wird vom **Smart Home-System** gesprochen, wenn das System, bestehend aus allen elektrischen Geräten im Haushalt, zentral und adaptiv im Sinne des Verbrauchers sowie des minimalen Energieverbrauchs gesteuert wird, ohne, dass ein Nutzereingriff nötig ist.

Definition 3. Smart Home-System.

Die nächst höhere autonome Abstraktionsebene nach adaptiv ist nach IBM **manageable** (kontrollierbares), danach kommt **self-manageable** (sich selbst kontrollierend), **self-managing** (sich selbst regelnd) und als höchste Abstraktionsebene **self-organizing** (sich selbst organisierend) [90]. Dabei gibt es fließende Übergänge zwischen jeweils benachbarten Systemabstraktionen [92]. Diese letzten vier Systemreifegrade sind der Vollständigkeit halber aufgezählt, spielen bei der vorliegenden Systementwicklung jedoch keine Rolle.

2.3.2 Grundsätzliche Topologien lernender Systeme

2.3.2.1 Überwachtes Lernen

Wird ein Modell des Lehrers als lernendes System erstellt und optimiert, spricht man vom überwachten Lernen (**supervised learning**) [4], dessen Topologie in Abbildung 12 gezeigt ist.

Die Modellbildung erfolgt, indem sowohl Lehrer als auch Modell zu einem bestimmten Zeitpunkt t ($t = 0, \dots, M-1$) mit denselben Eingangsgrößen, zusammengefasst im n -dimensionalen Vektor $x(t)$, konfrontiert werden (s. (3)). Dieser besteht beispielsweise aus Umweltinformationen wie der Uhrzeit, der Sonneneinstrahlung und der Luftfeuchtigkeit als skalare Größen.

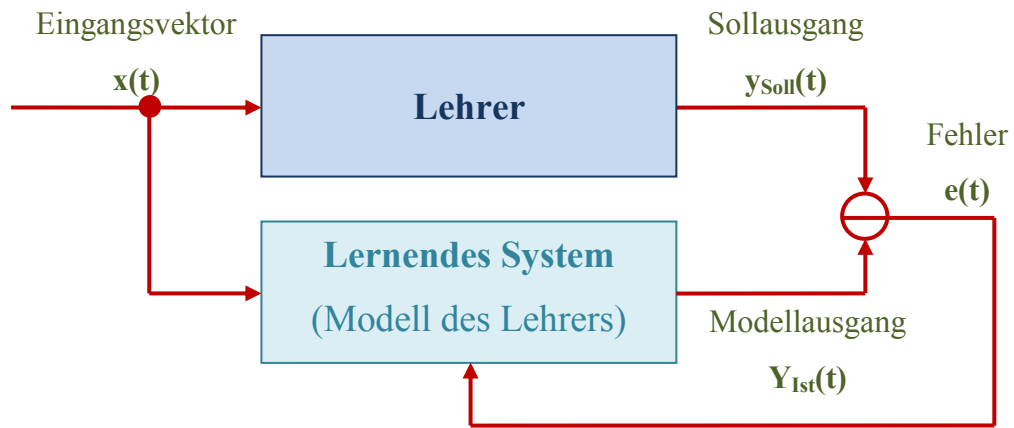


Abbildung 12. Blockschaltbild überwachtes Lernen.

Als Reaktion darauf werden sowohl der Lehrer als auch das Modell eine Antwort als m -dimensionale Vektoren $y_{\text{soll}}(t)$ bzw. $y_{\text{ist}}(t)$ bilden (s. (4) und (5)). Die Differenz beider Ausgänge ergibt die Abweichung des Modellausgangs vom Sollausgang bzw. den Fehlervektor, $e(t)$, der ebenfalls die Länge m hat und mit (6) berechnet wird. Das Modell wird mit $g(\mathbf{X}, \boldsymbol{\theta})$ beschrieben, wobei \mathbf{X} alle Vektoren $x(t)$ der Eingangsdaten und $\boldsymbol{\theta}$ den aktuellen Parametersatz beschreibt.

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \dots \\ x_n(t) \end{bmatrix} \quad (3)$$

$$y_{\text{soll}}(t) = \begin{bmatrix} y_{\text{soll},1}(t) \\ y_{\text{soll},2}(t) \\ \dots \\ y_{\text{soll},n}(t) \end{bmatrix} \quad (4)$$

$$\mathbf{y}_{\text{ist}}(t) = \begin{bmatrix} y_{\text{ist},1}(t) \\ y_{\text{ist},2}(t) \\ \dots \\ y_{\text{ist},n}(t) \end{bmatrix} \quad (5)$$

Beim „Lehrer“ bzw. der Umwelt, die modelliert werden soll, liegt dabei ein (intrinsisches) Strukturmodell zugrunde, das durch das Modell inklusive dessen Parameter $g(\mathbf{X}, \Theta)$ abgebildet werden soll, indem der Fehlervektor in das System zurückgeführt und zur Parameterkorrektur ausgewertet wird.

$$\mathbf{e}(t) = \mathbf{y}_{\text{soll}}(t) - \mathbf{y}_{\text{ist}}(t) = \begin{bmatrix} y_{\text{soll},1}(t) - y_{\text{ist},1}(t) \\ y_{\text{soll},2}(t) - y_{\text{ist},2}(t) \\ \dots \\ y_{\text{soll},m}(t) - y_{\text{ist},m}(t) \end{bmatrix} \quad (6)$$

Alle zur Verfügung stehenden Datensätze, bestehend aus Eingangs- und dazugehörigen Ausgangsvektoren, werden beim überwachten Lernen in zwei beziehungsweise drei Teilmengen gespalten, einerseits N Datensätze als Menge \mathbf{X} zum Belehren des Systems ($N < M$) und $M-N$ Datensätze zur Validierung (sowie dem Test) des Modells als Menge \mathbf{V} (s. (7) und (8))⁵.

$$\mathbf{X} = \{x(t), y_{\text{soll}}(t)\}_{t=0}^{N-1} \quad (7)$$

⁵ Häufig wird die Menge der Validierungsdatensätze in die eigentliche Validierung sowie den Testdatensätzen gegliedert. Da diese Unterscheidung in der vorliegenden Arbeit keinen wesentlichen Einfluss auf das Ergebnis hat, werden beide Mengen zukünftig zusammengefasst.

Jeder Datensatz wird dabei stochastisch der Trainings- oder Validierungsmenge zugeordnet. Diese Zuordnung kann sich auch verändern, wenn ein Trainings- und Validierungsvorgang durchgeführt ist und ein weiterer Durchlauf gestartet wird (Kreuzvalidierung).

$$\mathbf{V} = \{x(t), y_{soll}(t)\}_{t=N}^{M-1} \quad (8)$$

Beim Belehren dieses Modells werden nacheinander alle Eingangsdatensätze aus \mathbf{X} dem Modell präsentiert (Offline-Verfahren) und der jeweilige Fehlervektor $e(t)$ ($t = 0 \dots N-1$) als Rückmeldung ins System $g(\mathbf{X}, \Theta)$ eingespeist. Die Grundstruktur wie beispielsweise die Ordnung von $g(\mathbf{X}, \Theta)$ wird während dieser Phase in der Regel nicht mehr angepasst, sondern vorab definiert, die Koeffizienten werden allerdings permanent verändert, um den aktuell gemessenen Summenfehler $E(\mathbf{X}, \Theta)$ zu minimieren (s. (9)).

$$E(\mathbf{X}, \Theta) = \frac{\sum_{t=0}^{N-1} \|e(t)\|}{N} \quad \text{für die Modellkonfiguration } \Theta \quad (9)$$

Die anschließende Validierung wird das fertige strukturelle Modell inklusive deren Parameter bewerten, indem dieses mit den Validierungsdatensätzen \mathbf{V} beaufschlagt wird und aus allen erhaltenen Einzelfehlern (Betrag des Vektors $e(t)$) der Durchschnittswert gebildet wird, wie in (10) gezeigt ist.

$$E(\mathbf{V}, \Theta) = \frac{\sum_{t=N}^{M-1} \|e(t)\|}{N - M} \quad \text{für die Modellkonfiguration } \Theta \quad (10)$$

Je nach Wahl des Modellansatzes ist ein gewisses Potential der Umweltabbildung möglich (vergleichbar mit dem maximalen technischen Wirkungsgrad). Wenn zum Beispiel ein sehr komplexer Lehrer durch ein einfaches Modell nachgebildet werden soll, wird das zur Unteranpassung (**Underfitting**) führen, da auch durch noch so viele Lernzyklen kein gutes Ergebnis erzielbar ist. Eine Lösung ist in diesem Fall die Erhöhung der Komplexität des Modells.

Auf der anderen Seite ist eine Überanpassung (**Overfitting**) gegeben, wenn ein zu komplexes Modell gewählt wird. In diesem Fall wird das Rauschen, das – meist physikalisch gegeben – das Nutzsignal überlagert, mitgelernt und/ oder sich das Modell zu sehr an die präsentierten Trainingsdaten anpasst, womit die Fähigkeit zur Generalisierung verloren geht. Eine Abhilfe ist in diesem Fall die Vereinfachung des Modells, bis ein minimaler durchschnittlicher Fehler $E(\mathbf{V}, \Theta)$ erreicht wurde, sowie ein Trainingsabbruch, sobald sich der durchschnittliche Fehler für den Testdatensatz wieder vergrößert ($E(\mathbf{Y}, \Theta_t) > (E(\mathbf{Y}, \Theta_{t+1}))$).

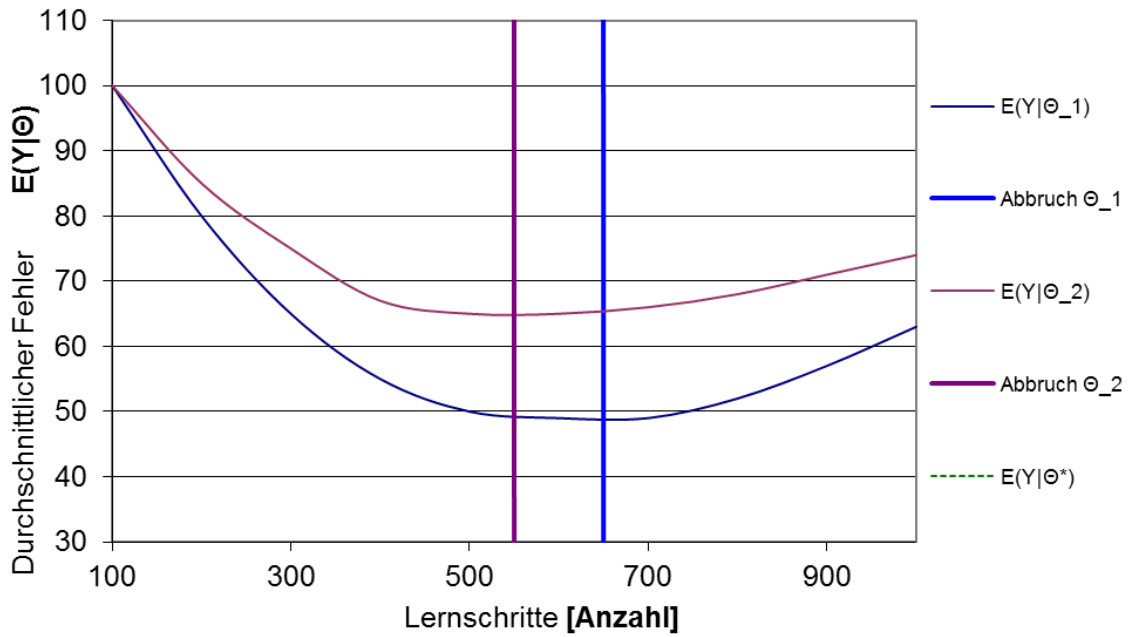


Abbildung 13. Entwicklung des durchschnittlichen Fehlers für zwei Modelle für dieselben Trainings- und Validierungssätze mit deren Abbruchdatensätzen. Für das Modell Θ_1 ist der minimale durchschnittliche Fehler $E(Y, \Theta_1)$ eher erreicht, aber größer als für das Modell Θ_2 . Das bedeutet zwar eine größere Komplexität des Modells Θ_2 und damit verbunden auch mehr Lernaufwand, aber auch eine bessere Modellierung des Lehrers.

Beispielsweise sollen in Abbildung 13 zwei unterschiedlich komplexe Modelle Θ_1 und Θ_2 das Verbraucherverhalten einer Person für ein Smart Home-System lernen. Da der minimale Fehler $E_{\min}(V, \Theta_1^*) < E_{\min}(V, \Theta_2^*)$ ist, dass der Modellansatz Θ_1 geeigneter (für die präsentierten Beispiele) ist als Θ_2 . Damit ist die Suchrichtung der Modellkonfiguration in Richtung Θ_1 sinnvoll, bis die optimale Parameterkonfiguration Θ^* gefunden wurde, was sich durch den (nicht abgebildeten) minimalen durchschnittlichen Fehler $E(V, \Theta^*)$ dokumentiert.

$$\Theta^* = \arg \min_{\Theta} E(Y, \Theta) \quad (11)$$

Die Berechnung dieser optimalen Konfiguration des Modells ist in (11) gezeigt. Zu beachten ist dabei, dass die zugrunde liegenden Validierungsdaten repräsentativ für das zu modellierende System, also Abtastpunkte und Auflösung entsprechend gewählt sein müssen, beispielsweise durch optimale Versuchsplanung. Desweiteren spielt der gemessene Zeitbereich eine wichtige Rolle, der ebenfalls hinreichend groß sein muss, damit alle Systemzustände, also beispielsweise ein komplettes zu messendes Jahr mit allen Jahreszeiten und verschiedenen Tages- und Nachtzuständen, im Smart Home, erfasst wurden.

In einem Smart Home-System ist der Ansatz des überwachten Lernens nicht sinnvoll, da der Anwender permanent aktive Rückmeldung über seine Zufriedenheit (als Lehrer) geben muss. Das ist nicht im Sinne eines Smart Home nach der Definition 3, das selbständig die Nutzerzufriedenheit ermitteln und daraus Rückschlüsse ziehen soll.

2.3.2.2 Bestärkendes Lernen

Ein zweiter Ansatz wird gewählt, wenn das Modell die Interaktion mit der Umwelt abbildet (**Reinforcement Learning**). Hierbei wird zunächst das lernende System mit den Eingangsgrößen, das ist im gezeigten Beispiel die zusammengefasste bestimmte Umweltsituation $s(t)$ zum Zeitpunkt t , beaufschlagt (s. Abbildung 14).

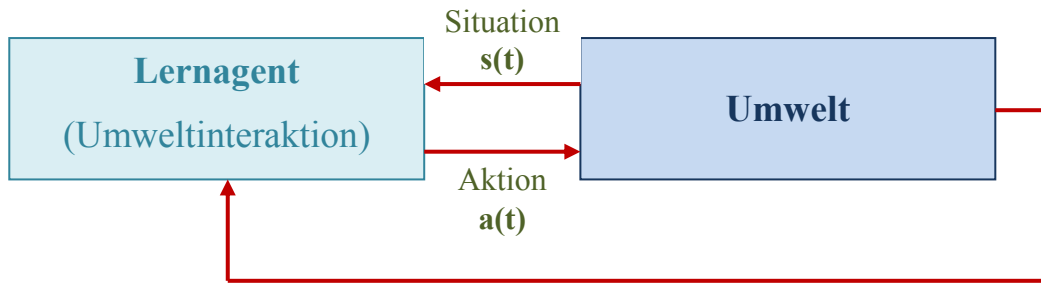


Abbildung 14. Blockschaltbild bestärkendes Lernen.

Der Vektor $s(t)$ setzt sich, wie in (12) ersichtlich, aus p skalaren Größen zusammen, die in der Regel messtechnisch erfasst und zur Verarbeitung aufbereitet werden. Anschließend wird der Lernagent mit einer Aktion $a(t)$ seine Umwelt soweit möglich steuern, um damit einem langfristigen Ziel näher zu kommen (z. B. Minimierung des Energieverbrauchs). Allerdings können nicht alle Größen der Umwelt beeinflusst werden, wie beispielsweise die Außentemperatur oder der aktuelle Niederschlag.

$$s(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \dots \\ s_p(t) \end{bmatrix} \quad (12)$$

Danach wird die Umwelt selbst, also der Anwender im Smart Home auf die durch eine Steuerung durchgeführte Strategie, auf die gewählte Aktion reagieren, indem sie von der Umgebung bestraft oder belohnt wird. Diese Antwort spiegelt sich in der skalaren Größe $r(t+1)$ wider. Die Zeit ist bei diesem Lernverfahren diskretisiert, wobei die Antwort von der Umwelt einen Zeitschritt später erfasst wird (damit die Umwelt Zeit für eine Reaktion hat).

Wichtig ist an dieser Stelle ist, die Belohnung (bzw. Bestrafung) der richtigen Aktion zuzuordnen. Denn die Antwort muss nicht unmittelbar nach der durchgeführten Aktion erfolgen und auch nicht am Stück, außerdem kann eine Antwort auf dieselbe Aktion, jedoch veränderter Vorgeschichte, durchaus anders ausfallen.

Deshalb wird an dieser Stelle die Markoveigenschaft erster Ordnung („Markovsches System“) definiert, bei dem der aktuelle Zustand ausschließlich vom unmittelbaren Vorzustand abhängt [39]. Gegebenenfalls müssen dafür Informationen aus vorangegangenen Zuständen in den direkt vorherigen integriert werden, damit sie nicht verloren gehen. Damit ist dafür gesorgt, dass sich die empfangene Umweltantwort $r(t+1)$ vollständig und ausschließlich auf die durchgeführte Aktion $a(t)$ ausgehend von der Situation $s(t)$ bezieht.

Ein Zustand s_t besitzt die **Markoveigenschaft**, wenn dieser nur vom direkten Vorzustand, jedoch nicht weiteren Vorzuständen abhängt, wie durch folgende Formel ausgedrückt wird:

$$\Psi(s_t | s_{t-1}) = \Psi(s_t | s_1, \dots, s_{t-1}),$$

Ψ ... Transition in den Zustand s_t

Definition 4. Markoveigenschaft eines Systems.

Um die optimale Steuerstrategie in einer bestimmten Situation $s(t)$ zu finden, wird der Gütewert $Q(a)$ eingeführt, also die (erfahrungsgemäße) erwartete Umweltreaktion, wenn die Aktion $a(t)$ gewählt wird, während die Situation $s(t)$ vorliegt. Zur Abbildung der Q-Werte wird üblicherweise eine Tabelle benutzt, die aus drei Spalten besteht: $s(t)$, $a(t)$ und $Q(a|s)$.

Um eine Aktion in einer bestimmten Situation zu wählen, wird nach der ε -greedy-Methode mit der Wahrscheinlichkeit von ε , $\varepsilon \in [0, 1]$, typischerweise im Bereich um $\varepsilon_{\text{typ}} = 0,9$, die nach aktuellem Wissensstand erfolgversprechendste Aktion $a^*(t)$ (s. (13)) gewählt (Exploitation). Mit der Restwahrscheinlichkeit $1 - \varepsilon$ bzw. wenn noch gar keine passende Aktion in der Wissensbasis vorhanden ist, wird eine beliebige Aktion durchgeführt und somit eine unbekannte Aktion ausprobiert (Exploration).

$$Q(a^*) = \max_a (Q(a)) \quad (13)$$

Im ersten Fall wird, nachdem die Umweltantwort $r_a(t+1)$ auf die gewählte Aktion erhalten wurde, nach der Deltaregel in (14) der Q-Wert aktualisiert. Falls, wie im zweiten Fall, noch kein Q-Wert vorliegt, wird dieser direkt als $Q_{t+1}(a) = r_a(t+1)$ aus der Umweltantwort übernommen.

$$Q_{t+1}(a) = Q_t(a) + \eta[r_{t+1}(a) - Q_t(a)] \quad (14)$$

Die Lernrate η beschreibt den Einfluss der Differenz zwischen dem vorher gelernten Q-Wert und der aktuell erhaltenen Umweltantwort. Je größer η ist, desto mehr wird mit $r_{t+1}(a)$ die aktuelle Antwort den Q-Wert verändern. Typischerweise wird η im Bereich $0,05 \dots 0,2$ gewählt bzw. permanent verringert (umso größer der Reifegrad der Wissensbasis wird).

Neben der Rückmeldung an den Agenten hat sich ein anderer Umweltzustand eingestellt, der als Situation $s(t')$ bezeichnet wird (und gleichzeitig den Ausgangszustand für den nächsten Lernzyklus darstellt). Das bestärkende Lernen findet im technischen Bereich in der Regelungstechnik die Hauptanwendung. Aufgrund der Möglichkeit, online zu lernen und die gelernten

Informationen direkt wieder zu nutzen, ergeben sich bereits nach wenigen Zyklen gültige Ergebnisse. Damit sind nicht erst sehr viele Daten zu erheben, das (abstrakte) Modell bildet sich in der Q-Werttabelle.

Bestärkendes Lernen wird jedoch meist nicht für singuläre Entscheidungen eingesetzt, um ein Problem zu lösen, weil das für viele Probleme, beispielsweise der Steuerung komplexer Systeme, nicht möglich ist. Stattdessen liegt die eigentliche Stärke dieses Lernverfahrens im Finden einer Steuerstrategie $\Pi(\bullet)$ als Folge (Vektor) von Steuereingriffen, um ein bestimmtes Ziel schlussendlich zu erreichen⁶. Am Beispiel eines Smart Home, bei der erst nach einer komplexen Folge von Aktionen das Ergebnis vorliegt, wenn eine bestimmte Steuerstrategie $\Pi_{\text{gut}}(\bullet)$ zur Lösung einer bestimmten Aufgabe geführt haben sollte. Dabei führt die Steuerstrategie $\Pi^*(\bullet)$ zur bestmöglichen Lösung, also diejenige Lösung, die mit minimalen Kosten (bzw. maximaler Anwenderzufriedenheit) zum Ziel führt.

Der Aufwand selbst, wenn mit Zustand s_t begonnen und die Steuerstrategie $\Pi(\bullet)$ benutzt wird, wird als Nutzen $V^\Pi(s_t)$ bezeichnet. Dabei sind alle erwarteten Belohnungen (bzw. Bestrafungen) kumulativ addiert, die bis zur Erreichung des Ziels erwartet werden.

Je nach vorliegender Umwelt gibt es zwei Ansätze zur Berechnung des Nutzens. Zum einen können eine endliche Anzahl von Aktionen in zeitdiskreten Abständen zu einer Episode zusammengefasst werden, wenn die Lösungswege zeitlich konkret eingegrenzt werden können, wofür die Berechnung des Nutzens in (15) für den Zeitbereich $1 \dots T$ dargestellt ist. Beispielsweise sollen in einem Smart Home Lampen über einen Tag hinweg so gesteuert

⁶ Solange die Parameter von Funktionen noch nicht geklärt sind, wird (\bullet) verwendet, um diese Funktion prinzipiell zu erläutern.

werden, dass die Anwender möglichst wenig enttäuscht werden und keine eigenen Nutzereingriffe durchführen müssen. Dabei gibt es einen definierten Anfangs- und Endpunkt (z. B. 0:00 bis 24:00 Uhr).

$$V^{\Pi}(s_t) = E \left[\sum_{i=1}^T r_{t+i} \right] \quad (15)$$

Im Gegensatz zu diesem Ansatz gibt es auch Probleme, deren Lösung eine sehr große bzw. unendlich große Anzahl an Aktionen verlangt (also das Ziel nie erreicht, sondern nur angenähert werden kann), wie beispielsweise verhaltenspsychologische Fragestellungen, deren Entwicklungen Jahre vorher beginnen. In diesem Fall können unmöglich erwarteten Belohnungen mit vertretbarem Aufwand gesucht werden. Dann wird ein gewisser Zeithorizont für die Nutzenberechnung herangezogen, indem die früheren Belohnungen, wie in (16) gezeigt, stärker gewichtet werden als die späteren. Damit konvergiert der rechte Term in (16) gegen die richtige Lösung in endlicher Zeit.

$$V^{\Pi}(s_t) = E \left[\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i} \right] \quad (16)$$

γ ($\gamma \in [0, 1]$) ist dabei die Diskontierung, die in (16) exponentiell gewichtet wird. Der Wert wird in der Regel deutlich kleiner als 1 gewählt, um aktuelle Belohnungen stärker zu bewerten. Die optimale Strategie $V^*(s_t)$ ist in (17) ausgedrückt.

$$V^*(s_t) = \max V(s_t) \quad \forall s_t \quad (17)$$

Ein wesentliches Problem dieses Ansatzes ist es, dass die Anzahl der möglichen Zustände kombiniert mit allen möglichen Aktionen eine sehr große Anzahl von Paarungen ergibt. Jede dieser Kombinationen müsste mehrfach durchlaufen werden, damit eine repräsentative zu erwartende Belohnung mit einer bestimmten Aussagesicherheit in der Wissensbasis abrufbar ist. Außerdem sind viele Eingangsparameter kontinuierlich und werden erst durch die Messung quantifiziert. Damit könnten aufgrund der Messtoleranzen zwei gleiche Zustände verschiedenen s_t -Werten zugeschlagen werden.

Die Lösung besteht in der Generalisierung von Gütewerten von (s, a) -Paaren. Ähnlich wie in Neuronalen Netzen oder anderen überwachten Lernverfahren, kann das Bestärkende Lernen ebenfalls als Regressionsmethode interpretiert werden. Dieser Ansatz wird später bei der Anwendung von Learning Classifier Systemen (Abschnitt 2.3.4) erfolgreich eingesetzt, da sich insbesondere bei Smart Home-Systemen bestärkendes Lernen sehr gut eignet, um die Verhaltensweisen der Anwender zu adaptieren. Hierbei ist keine direkte Programmierung des Smart Home-Systems nötig, was gleichzeitig eine der Anforderungen in der vorliegenden Arbeit ist.

2.3.2.3 Unüberwachtes Lernen

Das unüberwachte Lernen (**unsupervised learning**) besitzt, im Gegensatz zu den vorherigen Lernmethoden, keinen Lehrer, sondern wird in erster Linie zur Klassifizierung von Eingangsdaten benutzt. Dabei werden üblicherweise Gesetzmäßigkeiten in den Daten gesucht, die als Informationsquelle dienen. Dabei entsteht ein Datenmodell, das diese generelle Gesetzmäßigkeiten in Eingangsdaten $x(t)$ modelliert. In der vorliegenden Arbeit kann beispielsweise unüberwachtes Lernen eingesetzt werden, um allgemeine Ge-

setzmäßigkeiten über menschliche Handlungsweisen zu lernen, ohne, dass der Mensch selbst die Entscheidung des Modells direkt bewertet.

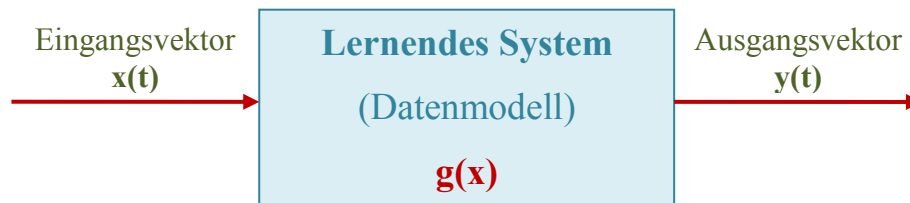


Abbildung 15. Blockschaltbild unüberwachtes Lernen.

Das Blockschaltbild der Modellbildung ist in Abbildung 15 gezeigt. Dabei wird das lernende System mit dem Eingangsvektor $x(t)$ beaufschlagt, während intern die Struktur der Daten durch die Funktion $g(x)$ charakterisiert wird. Die Eingangsdaten sind typischerweise in technischen Systemen Sensorinformationen, die eine Summe aus deterministischen und stochastischen Informationen liefern. Die Funktion $g(x)$ hat deshalb in der Regel einen stochastischen Anteil $g_{\text{stoch}}(x)$ und einen deterministischen $g_{\text{det}}(x)$. Der Ausgang $y(t)$ ist dann entweder die Information über die Eingangsdaten selbst, wie beispielsweise statistische Verteilungsparameter, oder bereits eine Entscheidung über die Gültigkeit der Eingangsinformationen. Daneben könnte auch eine Verarbeitung von $x(t)$ erfolgen und entsprechend ausgegeben werden.

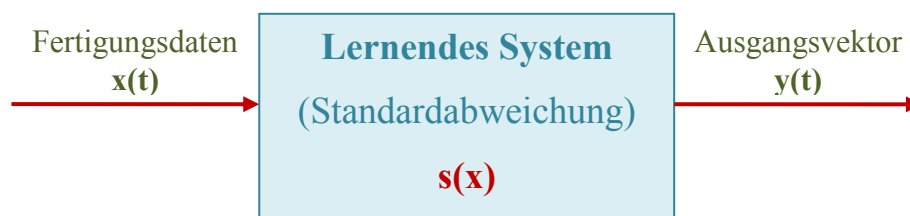


Abbildung 16. Unüberwachtes Lernen im Anwendungsbereich (Überwachung von Prozessen in der Fertigung).

Eine Anwendung für die Überwachung (*monitoring*) wäre ein Fertigungsprozessschritt, bei dem ein Parameter permanent erfasst und in seinen deterministischen $s_{\text{det}}(t)$ sowie seinen stochastischen Anteil $s_{\text{stoch}}(t)$ zerlegt wird (s. Abbildung 16). Handelt es sich bei $s_{\text{det}}(t)$ um eine Normalverteilung, sind Fertigungsfehler gut durch die Standardabweichung $s(x) = \sigma(s_{\text{det}}(t))$ charakterisierbar. Werkzeugverschleiß, einen Sensordefekt oder den Einsatz falscher Halbzeuge erkennt man dann, wenn sich $s(x)$ plötzlich für eine standardisierte Stichprobe (z. B. Ausstoß einer Schicht) gegenüber der Referenz $s(x')$ als Gutteile signifikant ändert.

Weitere Anwendungsbereiche sind Komprimierungsverfahren (beispielsweise zum Farbmapping von Bildern) sowie die Klassifizierung von Eingangsdaten sowie die Datenkompression während der Internetkommunikation.

2.3.3 Künstliche Neuronale Netze als Methode zur Verhaltensadaptierung

Eine Möglichkeit, sowohl die Verhaltensweisen der Benutzer im Haushalt als auch die elektrische Energieaufnahme der Geräte in Abhängigkeit von den Umweltbedingungen zu lernen, ist der Einsatz von Künstlichen Neuronalen Netzen. Diese können prinzipiell als überwachte, unüberwachte oder als bestärkende Lernmethode eingesetzt werden, je nach Topologie des Gesamtsystems. Desweiteren sind Künstliche Neuronale Netze fähig, stark nicht lineare Zusammenhänge abzubilden, wie beispielsweise die menschliche Entscheidungsfindung, die von vielen Einflussgrößen abhängt.

Im Rahmen der vorliegenden Arbeit wurden mehrere Topologien von Künstlichen Neuronalen Netzen im Abschnitt 6.3.1 ausprobiert, um Vorhersagen über Verbrauchervünsche für ein Smart Home-System zu treffen. Die

Konfiguration der Künstlichen Neuronalen Netze wurde dabei mit Hilfe eines Evolutionären Algorithmus‘ durchgeführt mit dem Ziel, die optimale Anzahl an Schichten und Neuronen für das Vorhersageproblem zu finden. Weitere Details über Künstliche Neuronale Netze sind im Anhang C genannt.

2.3.4 Learning Classifier Systems

Learning Classifier Systems (LCS) werden als zweite Lernmethode vorgestellt, da sie, wie nachfolgend gezeigt wird, ebenfalls als Lösungsmethode in Betracht kommen. Sie sind eine Kombination aus regelbasierten, kognitiven und adaptiven Methoden des Wissenserwerbs und finden hauptsächlich als bestärkende Lernverfahren Anwendung (es gibt aber auch durchaus Anwendungen, in denen LCS auch als überwachte Methoden verwandt werden) [88]. Im Gegensatz zu konventionellen Classifier Systemen wird hier die Regelbasis nicht manuell durch einen menschlichen Experten programmiert, sondern automatisch (maschinell) durch präsentierte Beispiele aus der Umwelt generiert und permanent gepflegt. Erstmals definiert wurde diese Methode durch J. H. Holland [52] [53] [54]. Neben den Einflüssen aus der dynamischen Programmierung [10] sind auch starke Einflüsse der Evolutionsbiologie auf die ersten LCS sichtbar [125].

Die Regelbasis (oder Population [**P**] nach [52]) bildet den Kern des LCS und wird durch die Beobachtung der Umwelt, in einem Haushalt beispielsweise die Benutzung von elektrischen Geräten durch die Anwender, und Interaktion mit dieser aufgebaut und permanent aktualisiert (s. Abbildung 17). Sie kann damit als Modell der Umwelt (in der vorliegenden Arbeit also der menschlichen Verhaltensweisen) mit einem bestimmten Reifegrad interpre-

tiert werden. Dieser Reifegrad ist umso größer, je kleiner dessen Vorher-
sagefehler in Bezug auf die menschlichen Verhaltensweisen ist, und umso
größere dessen Fähigkeit zur Generalisierung, also der generellen Gesetze
menschlichen Verhaltens, ist. Das Ziel ist die permanente Steigerung dieses
Reifegrads.

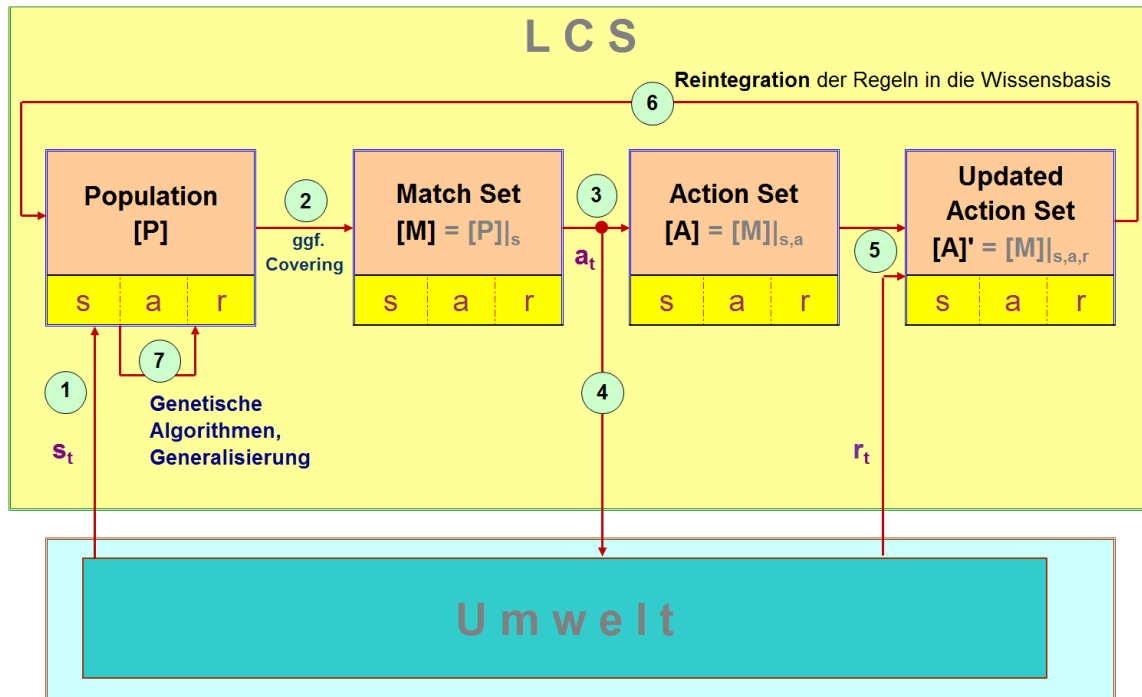


Abbildung 17. Systemtopologie eines LCS (eigene Darstellung).

Der prinzipielle Ablauf eines Lernzyklus‘ im LCS mit der Nummerierung der Schritte ist in Abbildung 17 gezeigt. Im ersten Schritt wird die Wissensbasis [P] mit einer bestimmten Situation s_t beaufschlagt, indem ein willkürlicher, repräsentativer und aktuell detektierter Umweltzustand zu einem bestimmten Triggerereignis (beispielsweise die Änderung einer Eingangsgröße) gelesen wird. Als Schritt 2 werden nun aus [P] alle Regeln in das Match

Set $[M]$ übernommen, zu denen diese Umweltsituation passt⁷. Damit ist zukünftig sichergestellt, dass nur von der Umwelt wirklich affektierte Regeln in den weiteren Schritten modifiziert werden.

Danach schlagen alle ausgewählten Regeln in einem Wettbewerb eine bestimmte Aktion a_t vor, die das LCS jetzt ausführen soll, um die Umwelt im Sinne einer langfristig sinnvollen Strategie zur Kostenminimierung zu beeinflussen. Dieser dritte Schritt mündet darin, dass aus $[M]$ wiederum eine bestimmte Anzahl von Regeln ausgewählt wird, die genau diese Aktion a_t vorschlagen und ins Action Set $[A]$ kopiert werden. Dabei wird in der Regel das Roulettewahlverfahren angewandt, da es sich in den allermeisten Problemen der Vergangenheit als besonders robust hinsichtlich der besten Lösung gezeigt hat [21]. Allerdings können bei speziellen Aufgabenstellungen auch andere Ansätze zum Einsatz kommen, die in Tabelle 1 dargestellt sind [63].

Ebenfalls wichtig für die Auswahl des besten Verfahrens im Schritt 3 ist, ob bei jedem Durchlauf alle Regeln wiederum mit den gleichen Bedingungen zur Selektion freigegeben sind (nicht diskriminierend) oder, ob vergangene Durchläufe einen Einfluss auf die aktuelle Auswahl haben. Ein Beispiel für letzteres ist das Eliteverfahren, bei dem nur diejenigen Regeln, die sich in vorherigen Selektionen sehr bewährt haben, an späteren teilnehmen. Normalerweise jedoch wird der erste, nicht diskriminierende Weg gewählt, um möglichst viele Arbeitsbereiche im Lösungsraum abzutasten.

⁷ Das ist dadurch ausgedrückt, dass $[M]$ auch als $[P]_s$ bezeichnet wird, also als Unter-
menge von $[P]$, die alle Regeln enthält, die zu s_t passen.

Selektionsmethode	Beschreibung
Zufällige Auswahl (Stochastic universal sampling)	Hierbei wird eine Regel aus $[M]$ rein zufällig ausgewählt (gleichverteilt)
Deterministische Auswahl	Die Regel mit dem größten zu erwartenden Gewinn wird gewählt, womit immer die stärkste Regel gewinnt und niemals eine Nischenaktion durchgeführt wird
Proportional (Rouletterad-Verfahren)	Alle Regeln werden – wie bei einem Glücksrad mit verschiedenen großen Segmenten – gemäß ihres zu erwartenden Gewinns gewichtet und danach eine Regel gewählt; die größer gewichteten Regeln haben dabei Vorteile, ausgewählt zu werden
Wettkampfselektion	Eine bestimmte Anzahl von Regeln wird zufällig ausgewählt (Schritt 1), von denen wiederum die mit dem stärksten zu erwartenden Gewinn angewandt wird (Schritt 2)
Boltzmann-Selektion	Hier wird der Selektionsdruck mit fortschreitender Modellreife immer weiter vergrößert, sodass mit jedem Lernzyklus weniger Lösungen zur Auswahl stehen [87]
Lineares Ranking	Hierbei wird nicht von der Stärke der Regeln (erwartete Belohnung durch die Umwelt) ausgegangen, sondern jede Regel mit einem festen Abstand hinter die nächst bessere platziert.

Tabelle 1. Methoden zur Auswahl einer Regel aus dem Match Set $[M]$ für das Action Set $[A]$. Die am häufigsten benutzte Methode des Rouletteradverfahrens ist türkis hinterlegt.

Gleichzeitig mit der Bildung von $[A]$ wird die Umwelt im vierten Schritt mit der gewählten Aktion a_t beaufschlagt. Damit wird die Umwelt in den Ar-

beitspunkt gebracht, den das LCS als den aussichtsreichsten hält. Bevor die Reaktion der Umwelt auf diesen Eingriff im sechsten Schritt erfasst und zurückübertragen wird, werden alle Regeln als Schritt fünf aus dem Action Set **[A]** in ein Updated Action Set **[A]'** übertragen, damit die Umweltantwort je nach Einfluss der Regeln auf die getroffene Steuerentscheidung zum Update (und damit zur Steigerung des Modellreifegrades) beitragen kann. Im nächsten Schritt werden die aktualisierten Regeln zurück in die Wissensbasis übertragen, womit die Erneuerung des Wissens infolge der Interaktion mit der Umwelt (Reinforcement) beendet ist.

Bei der Anzahl der in einem Lernzyklus (Schritte 1 bis 7 in Abbildung 17 nacheinander bilden einen Lernzyklus) veränderten Regeln der Population **[P]** wird zwischen der Generations- und der stationären Reproduktion unterschieden. Während bei der Generationsreproduktion alle Regeln der Population **[P]** einer Generation im Rahmen des Updates ersetzt werden, sind bei der stationären nur ein Teil der Regeln der Aktualisierung während eines Lernzyklus' unterzogen [16]. Im vorgestellten LCS werden per Definition nur diejenigen Regeln aktualisiert, die zur eingelesenen Umweltsituation s_t passen und gleichzeitig die gewählte Aktion vorschlagen⁸, deshalb spricht man hier von der stationären Reproduktion [56].

Anschließend folgen im 8. Schritt (vgl. Abbildung 17) die Genetischen Algorithmen [38], um die Wissensbasis selbst weiterzuentwickeln: Selektion, Rekombination und Mutation sowie die Generalisierung sorgen für die Exploration und die Exploitation der Wissensbasis. Damit werden zum einen permanent neue Arbeitspunkte im Lösungsraum ausprobiert, die bislang noch nicht betrachtet wurden, aber auf Basis der bislang gewonnen Erfah-

⁸ Das sind alle Regeln des Action Sets **[A]**

rungen aussichtsreich erscheinen. Zum anderen sorgt die Generalisierung dafür, im hyperdimensionalen Lösungsraum generelle Abhängigkeiten zwischen Einflussgrößen und einer bestmöglichen Lösung zu finden, um damit die Komplexität entscheidend zu reduzieren. Im psychologischen Sinne wird eine Präferenzkonzentration angestrebt, denn Menschen treffen ihre Entscheidungen unter Zuhilfenahme einer sehr begrenzten Anzahl von Einflussgrößen, im regelungstechnischen Sinne eine Parameteridentifikation vorgenommen [30]. Das Ziel ist dabei, die wesentlichen, den anderen zu Grunde liegenden Wirkzusammenhänge zu finden und eine ausreichend gute Modellabstraktion zu erzeugen.

Der Vorhersagefehler berechnet sich durch eine bestimmte Belohnung (bzw. Bestrafung) als Antwort aus der Umwelt auf eine getroffene Entscheidung des LCS. Das Ziel des Lernverfahrens ist es, Entscheidungen in Abhängigkeit bestimmter Umweltzustände derart zu treffen, um die langfristigen Belohnungen an das System zu erhöhen (Maximierungsproblem) bzw. die langfristigen Bestrafungen zu reduzieren (Minimierungsproblem)⁹, wenn gleich bestehende Regeln aus der Population zur Problemlösung eingesetzt werden (vgl. (18)). Bessere Regeln erhalten bei einem Maximierungsproblem also auch eine größere Belohnung als schlechtere und generellere kommen häufiger zur Anwendung als speziellere¹⁰, wobei im Laufe des Lernvorgangs immer mehr Regeln gebildet werden sollten, die möglichst für große Arbeitsbereiche eine Lösung mit großem zu erwartenden Gewinn anbieten.

⁹ Zukünftig wird, der Übersichtlichkeit halber, nur das Maximierungsproblem betrachtet, da die Methoden dieselben sind und die meisten praktischen Aufgaben Maximierungsprobleme sind. Außerdem kann jedes Minimierungs- in ein Maximierungsproblem überführt werden.

¹⁰ Damit ist die Generalisierung als implizites Ziel in den Lernmethoden der Learning Classifier Systeme enthalten.

$$\max(E(\sum_{t=0}^{\infty} \gamma^t r_t)) \quad (18)$$

Die Wissensbasis **[P]** besteht dabei aus maximal i_{\max} Regeln¹¹, die mit Ψ_i bezeichnet werden, wobei i ($0 \leq i \leq i_{\max}$) den Index der Regel bezeichnet. Jede Regel i besteht wiederum aus einem Situationsteil $\Psi_{i.s}$, einer Aktion $\Psi_{i.a}$ sowie einer zu erwartenden Belohnung $\Psi_{i.r}$ (Reward) als skaliertes Wert. Typischerweise liegt i_{\max} im Bereich 20 ... 1000 und kann als die Ordnung des Modells interpretiert werden¹². Denn bei einer großen Anzahl an Regeln müssen wesentlich später Regeln überschrieben werden, was bei einer komplexen Umwelt und gleichzeitig vielen Lernbeispielen schnell zu einer guten Lösung konvergiert, ansonsten jedoch zu langen Lernzyklen mit schlechter Güte des Modells führt [52].

In der Situation $\Psi_{i.s}$ können zum einen reale Zahlenwerte, logische Werte oder ganze Zahlen bzw. Bereiche und zum anderen der Joker (oder Wildcard) „#“ für jeden einzelnen Term (also ein Zeichen im String) vorkommen. Der Joker steht dabei für ein beliebiges Zeichen. Mit Hilfe dessen können Bereiche (oder Hyperebenen) im Zustandsraum erzeugt werden und damit ist eine Generalisierung möglich. Ein mit Jokern versehener String bildet ein Schema und ist eine sehr mächtige Methode bei lernenden Algorithmen, wie in Abschnitt

¹¹ Das System startet mit der leeren Wissensbasis ($i = 0$) und inkrementiert diesen Index mit jedem neu angelegten Satz solange, bis die maximale Anzahl i_{\max} erreicht ist. Danach werden Regeln bei Bedarf nur noch überschrieben.

¹² Der direkte Vergleich ist allerdings nicht möglich, da jede Regel einen Arbeitsbereich abbildet, welcher selbst bereits höherer Ordnung sein kann.

Wenn beispielsweise in einer bestimmten Situation eine Regel, unabhängig vom Wert einer bestimmten Eingangsgröße, greifen soll, wird für genau diese Eingangsgröße ein Joker gesetzt. Damit wird diese Regel bei jedem beliebigen Wert dieses Eingangs ausgewählt, wenn der Rest der Regel auch gewählt würde.

2.3.4.1 Das Covering

Im Sonderfall, falls sich keine zu s_t passende Regel in der Population befindet (da beispielsweise eine Umweltsituation im Smart Home detektiert wurde, die so noch nicht in der Wissensbasis vorhanden ist, weil etwa noch nie während der bisherigen Systemlaufzeit so viele Menschen im Haushalt anwesend waren), muss eine Regel für die aktuelle Situation generiert werden, denn ein leeres **[M]** ist nicht zulässig. Diesen Prozess nennt man Covering (vgl. Abbildung 17, Schritt 2).

Da die Wissensbasis beim Covering bereits voll ist (maximal i_{\max} Regeln in **[P]** zulässig), muss eine Regel aus dieser überschrieben werden, damit die neue, gecoverte Regel Platz findet. Hier gibt es die Möglichkeit, die ähnlichste (*kleinste* Distanz zwischen $\Psi_{i.s}$ und s_t) zu ersetzen, die verschiedenste (*größte* Distanz zwischen $\Psi_{i.s}$ und s_t) zu ersetzen oder eine bereits vorhandene Regel derart zu generalisieren, dass sie die aktuelle Situation mit abbildet.

Differenzberechnung

Zur Ermittlung dieser Differenz (im Folgenden mit $\Delta(\bullet)$ abgekürzt) gibt es verschiedene Ansätze, von denen die wichtigsten hier vorgestellt werden. Die Differenzermittlung selbst spielt eine entscheidende Rolle für die Generierung von Regeln, da sie der Ausdruck dessen ist, wie der Situationsraum zu interpretieren ist [143]. Das Ziel ist, dass für ähnlich wahrgenommene Situationen, beispielsweise die Regeln ii und jj , $\Psi_{ii.s}$ und $\Psi_{jj.s}$, auch die berechnete Differenz sehr klein ist. Erst dann wird der Zustandsraum optimal durchsucht. Damit ergibt sich bereits eine Randbedingung: für zwei identische Situationen soll die Differenz $\Pi(\Psi_{ii.s}, \Psi_{jj.s})|_{ii=jj} := 0$ sein.

	Licht 1	Licht 2	Licht 3
Situation aus der Population $\Psi_{i.s}$	1	0	1
Gelesene Umweltsituation s_t	0	0	1
Zeichen unterschiedlich?	1	0	0
Hammingdistanz	1		

	Licht 1	Licht 2	Licht 3
Situation aus der Population $\Psi_{i.s}$	1	0	0
Gelesene Umweltsituation s_t	0	1	0
Zeichen unterschiedlich?	1	1	0
Hammingdistanz	2		

Abbildung 18. Zwei verschiedene Hammingdistanzen (Ansteuerung von drei Lampen).

Am häufigsten wird dabei die Hammingdistanz $hd(\bullet)$ benutzt, bei der in zwei gleichlangen Strings s_{ii} und s_{ij} die Anzahl der unterschiedlichen Zeichen ermittelt wird [78] [144]. Die Berechnung ist in

Abbildung 18 veranschaulicht, am Beispiel eines Haushalts jeweils drei Lampen angesteuert sind (Gelesene Umweltsituation s_t) und mit dem Situationsteil der Regel Ψ_i verglichen werden. Dabei wird im Modell der Umwelt unterstellt, dass die Differenz zwischen zwei Zeichen jeweils die gleichen Kosten (beispielsweise in Folge einer Fehlsteuerung) verursachen würde, wenn sie auftritt. Deshalb ist diese Berechnung nur sinnvoll für binäre Zeichen, wobei jede Information prinzipiell in eine binäre Zahl umgewandelt werden kann (Ansatz beim Standard-LCS).

Der große Nachteil dieser Berechnungsmethode ist die meist zu einfache Abbildung der Umwelt. Im Smart Home wird beispielsweise der Verschleiß durch häufiges Ein- und Ausschalten von Geräten nicht mit berücksichtigt, der LEDs anders ist als bei Glühlampen. Dann wäre die Hammingdistanz ein zu einfacher Ansatz und würde dazu führen, dass ein zu häufiges Schalten nicht ausreichend bestraft wird. Ein Hauptanwendungsgebiet ist deshalb nicht Steuerung einfacher Geräte, sondern die Molekularbiologie [74].

Eine weitere Berechnungsmethode der Differenz $\Delta(\bullet)$ ist die Evolutionäre Distanz [130]. Bei diesem Ansatz werden die Kosten, die auftreten würden, wenn man einen String (z. B. „AG“) in einen anderen String (z. B. „T“) überführen würde, summiert. Am Beispiel einer DNA-Sequenz in Abbildung 19 bedeutet das, dass das Streichen und das Einfügen einer Säure in den DNA-Strang Kosten von 2 verursacht, wobei das Ersetzen einer Säure durch eine andere mit 3 bewertet wird. Wichtig ist, dass der kürzeste Weg der Überführungen angenommen wird. Um das zu erhalten, werden in [130] Regeln vorgeschlagen, in welcher Reihenfolge die Ersetz-, Einfüge- und

Streichoperationen vorzunehmen sind. Diese Methodik ist auch auf eine Zustandsänderung im Smart Home übertragbar, wenn beispielsweise vom Zustand „Alle Lichter ausgeschaltet“ in einen Zustand „Alle Lichter eingeschaltet“ sukzessive die Lampen nacheinander eingeschaltet werden. Dabei gibt es verschiedene mögliche Reihenfolgen, die Lampen zu schalten, wobei stets diejenige mit den wenigsten Schaltvorgängen gewählt werden soll.

	Kosten
$\Psi_{i,s} = \text{'AG'}$	
$s_t = \text{'T'}$	
Schritt 1: Entfernen 'A'	2
Schritt 2: Ersetzen 'G' durch 'T'	3
Evolutionäre Distanz	5

Abbildung 19. Berechnung der Evolutionären Distanz zwischen $\Psi_{i,s} = \text{'AG'}$ und $s_t = \text{'T'}$ als kürzeste Differenz zwischen $\Psi_{i,s}$ und s_t . Alternativ könnten die Operationen anders gewählt werden, womit sich die Distanz allerdings vergrößern würde, deshalb wurde auf alternative Berechnungen verzichtet.

Eine dritte Möglichkeit ist die Interpretation der einzelnen Parameter einer Situation als Informationen, die als Gesamtstring eine gewisse Entropie besitzen [23]. Diese Möglichkeit aus der Informationstheorie (Shannontheorie) spielt in der vorliegenden Arbeit keine Rolle und wird lediglich erwähnt.

Im Gegensatz zu allen bisherigen Theorien ist bei der L_1 - bzw. L_2 -Norm keine Umrechnung der realen Werte in binäre nötig. Dabei wird die Hammingdistanz durch den euklidischen Abstand (für L_2 -Norm quadriert) zwischen zwei Zeichen k_1 und k_2 aus einem String ersetzt. Daran angelehnt wird in typischen Zielfunktion einer Regelung der gewichtete Fehler zwischen Soll- und Istsignal (als Zielfunktion) als Berechnungsgrundlage herangezogen.

Integration einer per Covering generierten Regel in Wissensbasis:

Bei der Integration einer Regel Ψ_i in die Wissensbasis $[P]$ mittels Covering wird entweder die gelesene Situation s_t in die neue Regel i als $\Psi_{i.s}$ übernommen und wahlweise, wie in den meisten LCS üblich, zusätzlich noch eine zufällige Anzahl von Jokern eingesetzt (und die Regel damit bereits generalisiert) [141]. $\Psi_{i.a}$ und $\Psi_{i.r}$ werden dabei in der Regel wiederum von der ähnlichsten Regel (kleinste Hammingdistanz) übernommen, falls keine besseren Informationen wie bereits erhobene Sollwerte oder Vorwissen vorliegen sollten.

Nachdem i_{Match} ($1 \leq i_{Match} \leq i_{max}$) Regeln in $[M]$ kopiert wurden, wird eine Regel $\Psi_{selection}$ ausgewählt, die zur Anwendung kommen soll.

Nachdem die Auswahl stattgefunden hat, werden alle Regeln, die die Aktion der gewählten Regel $\Psi_{selection.a}$ ebenfalls vorschlagen würden, ins Action Set $[A]$ übernommen. Dieser dritte Schritt in Abbildung 17 dient dazu, alle gleichartigen Regeln in die spätere Belohnung (oder Bestrafung) seitens der Umwelt mit einzubeziehen. Gleichzeitig wird (als vierter Schritt) die Umwelt mit der Aktion beaufschlagt, wofür die Aktoren des Systems zuständig sind. Auch hier müssen ggf. Anpassungen des Datenformats vorgenommen und auf sichere Zustandsübergänge geachtet werden. Im Nachfolgenden wird davon ausgegangen, dass die getroffene Entscheidung vollständig umgesetzt werden kann (also die Hardwaretreiber korrekt arbeiten).

Im Smart Home bedeutet das, dass die Umweltsituation jetzt für nachfolgende Entscheidungen bekannt ist.

2.3.4.2 Der Reinforcementvorgang

Die Schritte 5 und 6 (s. Abbildung 17) beschreiben den Reinforcementvorgang selbst. Das bedeutet, dass die Regeln im Action Set belohnt (oder bestraft) werden, je nachdem, wie die Umwelt auf die ihr aufgezwungene Aktion $\Psi_{\text{selection}.a}$ reagiert. Bei der Spiegelung des Gewinns dieser Antwort auf die verwendeten Regeln haben sich zwei Verfahren durchgesetzt, der Bucket Brigade-Algorithmus (BBA) sowie das Q-Learning (als klassischer Ansatz aus dem bestärkenden Lernen) [75].

Für den BBA gibt es mehrere Möglichkeiten, die sich hinsichtlich der Verteilung des Gewinns etwas unterscheiden, wobei das Grundprinzip nachfolgend beschrieben wird [43]. Im ersten Schritt ist eine Auktion durchzuführen, bei der jede Regel aus $[A]$ ein Gebot entsprechend ihrer Spezifität sowie ihrer Stärke (hier vergleichbar mit dem zu erwartenden Gewinn $\Psi_i.r$ einer Regel i aus dem Action Set $[A]$) abgibt und anschließend in den Handel um den erhaltenen Gewinn r_t aus der Umwelt einbezogen wird.

Während der Auktion werden Gebote $B_t(i)$ für jede Regel in $[A]$ nach (19) berechnet. Die direkte Proportionalität des Gebots mit der Spezifität einer Regel $\text{ord}(\Psi_i)$ (Berechnung s. (28)) lässt sich dadurch erklären, dass eine abstraktere Regel, so wie in der Building Block-Hypothese beschrieben, auch einen deutlich größeren Bereich an Umweltzuständen abdeckt und damit anteilig für jede Regel nicht so relevant sein kann wie eine spezielle (Nischen-) Regel.

$$B_t(i) = k \bullet \Psi_i.r \bullet \text{ord}(\Psi_i) \quad (19)$$

Von allen Regeln aus **[P]** wird ein kleiner Wert von $\Psi_i.r$ abgezogen und diese Regeln damit pauschal geschwächt, damit sowohl Gewinne als auch Verluste möglich sind. Damit berechnet sich die neue Stärke anhand der (20).

$$\Psi_i.r(\text{neu}) = \Psi_i.r - \Psi_i.r \bullet \text{Tax}_{life} \quad (20)$$

Anschließend wird, wie in (21) gezeigt, von jeder Regel in **[A]** eine weitere kleine „Gebühr“ abgezogen, um überhaupt an der Auktion teilnehmen zu dürfen. Diese ist für jede Regel i proportional zu deren Gebot $B_t(i)$.

$$\Psi_i.r(\text{neu}) = \Psi_i.r - B_t(i) \bullet \text{Tax}_{life} \quad (21)$$

Im letzten Schritt wird die erhaltene Umweltantwort gelesen und an die Regeln ausgeschüttet, die die Auktion gewinnen, während sie ihre Gebote selbst verlieren (die allerdings bei jeder guten Regel geringer ausfallen sollten als die Umweltantwort). Die neue Stärke $\Psi_i.r(\text{neu})$ der Regel i setzt sich jetzt nun aus der alten Stärke $\Psi_i.r$ zusammen, von der das Gebot $B_t(i)$ abgezogen und sowohl der erhaltene Gewinn r_t als auch das mit γ diskontierte neue Gebot $B_{t+1}(i)$ hinzuaddiert werden, wie in (22) gezeigt ist.

$$\Psi_i.r(\text{neu}) = \Psi_i.r - B_t(i) + r_t + \gamma B_{t+1}(i) \quad (22)$$

Bei der Ausschüttung des Gewinns an Regeln, die nicht unmittelbar am Erfolg beteiligt sind, gibt es wiederum mehrere Ansätze, die zu erklären jedoch den Rahmen der vorliegenden Arbeit sprengen würden.

Eine häufig angewandte Modifikation dieses Algorithmus' ist das Implizite BBA (IBBA) [22]. Hier wird, im Gegensatz zur oben beschriebenen Varian-

te, jeder Regel aus $[A]$ der gleiche Anteil als Gebot abgezogen, unabhängig von deren derzeitigen Stärke (also der zu erwartenden Belohnung), wie in (23) gezeigt ist. Je nach vorliegendem Problem kann dieser pauschale Ansatz, der auch in Wilsons Zeroth Level Classifier System (ZCS) verwendet wird, schneller zu einer Lösung führen.

Dem Nachteil der starren Verteilung der Gebote steht der Vorteil der Lernrate β gegenüber, die dynamisch während des Lernvorgangs verringert werden kann, sobald sich die Wissensbasis einem gesättigten Zustand nähert. Damit ist ein schnelleres Konvergieren möglich.

$$\Psi_i \cdot r(\text{neu}) = \Psi_i \cdot r + \beta \left(\frac{r_t}{\| [A] \|} - \Psi_i \cdot r \right) \quad (23)$$

Im ZCS (Zeroth Level Classifier System) wird der Q-Lernalgorithmus eingesetzt, wie er in der frühen Phase der LCS angedacht war [126]. Damit wurde eine minimalistische, sehr effiziente Variante erzeugt, die jedoch nur für kleinere Probleme einsetzbar ist.

Die andere grundsätzliche und ebenfalls weit verbreitete Belohnungsstrategie wird vom Q-Learning übernommen. Sie ist allerdings nur für den Sonderfall XCS anwendbar [139]. Dabei wird jede Regel, wie weiter oben beschrieben, um die Komponenten in Tabelle 2 erweitert. Es werden, abhängig von der Lernrate β , alle Parameter einer Regel anhand der Rückmeldung aus der Umwelt gepflegt, wobei die Erfahrung permanent wächst.

Im Wesentlichen lehnt sich diese Art der Belohnung an der Widrow-Hoff-Regel (Delta-Regel), an die auf der Optimierungsmethode der minimalen Fehlerquadratsumme fußt und in (24) dargestellt ist [134].

$$x = x + \beta (y - x) \quad (24)$$

Die Lernrate β kann entweder statisch oder dynamisch (abnehmend abhängig vom Lernfortschritt) gewählt werden. Es sollen an dieser Stelle jedoch nicht alle Formeln zum Update einer Regel mit dem Q-Learning-Ansatz benutzt werden, sondern nur exemplarisch das Einfließen der erhaltenen Belohnung in (25).

$$\Psi_{i,r} = \Psi_{i,r} + \beta (r_t - \Psi_{i,r}) \quad (25)$$

Dabei kann der wirklich erhaltene Gewinn r_t als Q-Wert in der Q-Wertetabelle interpretiert werden.

Im siebten Schritt in Abbildung 17 werden die Regeln aus **[A]**' zurück in die Population **[P]** geschrieben und sowohl **[M]** als auch **[A]** und **[A]**' gelöscht. Damit ist sichergestellt, dass diese Untermengen der Regelbasis für den nächsten Zyklus initialisiert sind.

2.3.4.3 Pittsburgh- vs. Michigantyp

Für die Interpretation einer Regel an sich gibt es zwei verschiedene Ansätze, und zwar den Pittsburgh- und den Michigantyp. Beim Pittsburgh-Typ bildet jede Regel in der Wissensbasis eine Lösung des vorliegenden Problems ab, beispielsweise die Schaltmuster sämtlicher Geräte in einem Smart Home über einen ganzen Tag hinweg. Im Michigantyp hingegen wird nur ein logisch begrenzter Schritt in einer Regel kodiert, also die Herstellung eines bestimmten Schaltzustands der Geräte, sodass mehrere Regeln nacheinander angewandt werden müssen, um ein Ziel zu erreichen. Veranschaulicht ist

dies in Abbildung 20, wo in einem Haushalt als Beispiel die Schaltung aller Geräte über einen ganzen Tag hinweg betrachtet werden.

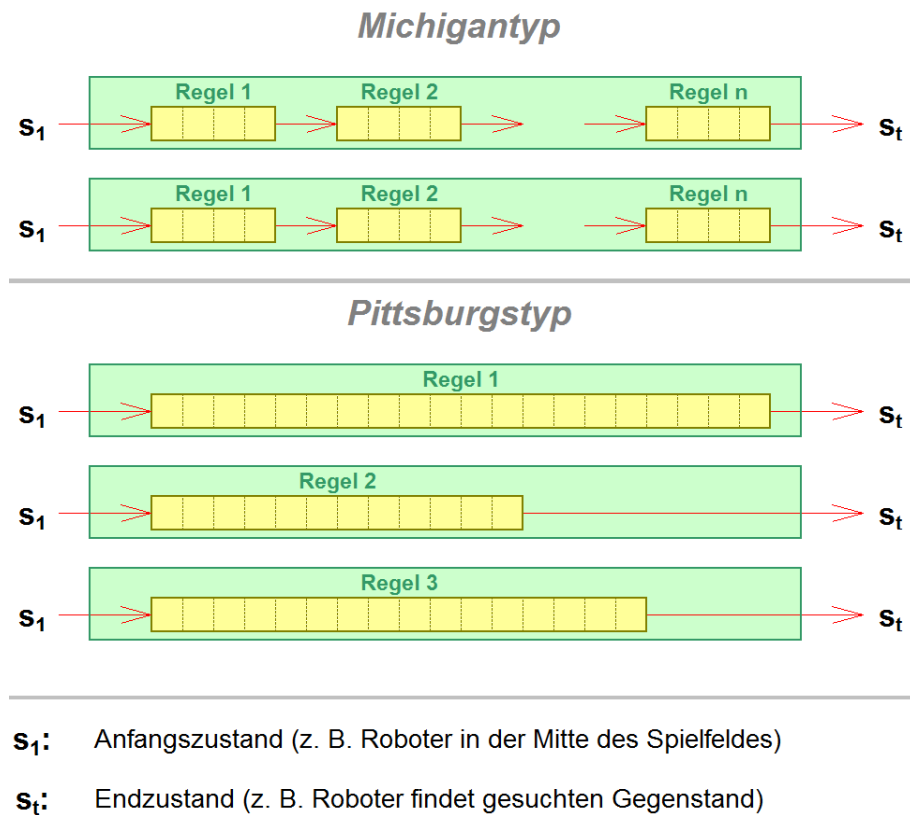


Abbildung 20. Gegenüberstellung von Pittsburgh- und Michigantyp in Bezug auf Regeln in der Wissensbasis.

2.3.4.4 Weiterentwicklungen und Modifikationen von LCS

Zahlreiche Modifikationen sowie Erweiterungen wurden in der Literatur untersucht, die bekannteste darunter ist das XCS. Hier wird, im Gegensatz zum LCS, nicht die Höhe der zu erwartenden Belohnung als Entscheidungsgrundlage benutzt, sondern die Genauigkeit der Vorhersage von Belohnungen, die Erfahrungen einer Regel sowie deren Fitness in einer Generation. Diese Änderungen waren eine Antwort auf die Probleme klassischer LCS, in denen verzögerte Antworten seitens der Umwelt oft zu Übergeneralisierung

sowie sehr ungleichmäßiger Verteilung von Regeln im Problemraum führten [140]. Dafür werden alle Regeln neben der Gewinnvorhersage $\Psi_{i,r}$ noch um die Komponenten in Tabelle 2 erweitert. Desweiteren modifiziert der genetische Algorithmus die Regeln nicht in der Population, sondern löscht diese nur.

Komponente	Formelzeichen der Regel i	Beschreibung
Vorhersagefehler/ Prognosegenauigkeit	$\Psi_{i,\epsilon}$	Durchschnittlicher absoluter Fehler bisheriger Vorhersagen dieser Regel
Fitness	$\Psi_{i,F}$	Genauigkeit der Regel relativ zu anderen
Erfahrung	$\Psi_{i,E}$	Messung des Erfolgs der Regel

Tabelle 2. Nötige Erweiterungen der Regeln eines LCS, um ein XCS zu schaffen.

Damit kann ein XCS-basierter Ansatz (auch als stärkebasierter Ansatz bezeichnet) eher bei Aufgabenstellungen zum Einsatz kommen, in denen Nischenlösungen gefragt sind und findet seit Ende der 90er Jahre immer mehr Verbreitung [20] [81] [55]. Vor allem die Messung des Erfolgs jeder Regel, der sich mit jedem erfolgreichen Einsatz der Regel verbessert und umgekehrt, hat sich als nützlicher Ansatz erwiesen [137]. Den LCS hat dieser Ansatz damit zu neuem Schwung verholfen, indem neue Anwendungsmöglichkeiten erschlossen wurden.

Ein weiteres Merkmal von XCS ist das Bestreben, einerseits Regeln in verschiedenen Nischen möglichst gleichverteilt einzusetzen, und andererseits möglichst alle Nischen im vorhandenen Lösungsraum zu besetzen. Das wird durch den Algorithmus zum Löschen von Regeln realisiert, sobald die zuläs-

sige Größe der Regelbasis überschritten ist. Dabei wird die Löschwahrscheinlichkeit einer Regel umgekehrt proportional zu deren Fitness und deren Erfahrung berechnet. Denn eine unerfahrene Regel mit geringer Fitness sollte prinzipiell eher aus der Wissensbasis entfernt werden als eine erfahrene mit großer Fitness. Allerdings sind auch hier – wie bei der Auswahl der Aktion – andere Möglichkeiten denkbar. Zum einen die Roulette Wheel-Selektion mit der umgekehrten Fitness als Auswahlwahrscheinlichkeit und zum anderen die rein zufällige, nicht gewichtete Auswahl von zu löschenden Regeln. Diese beiden Varianten bilden die Umwelt nicht hinreichend ab und werden deshalb nicht eingesetzt.

Eine weitere Möglichkeit ist, den Regeln eine Evolution aufzuprägen. Das bedeutet, dass „ältere“ Regeln tendenziell häufiger gelöscht werden als jüngere. Nach der maximalen Lebensdauer werden sie in jedem Fall entfernt. Diese Variante scheidet allerdings auch aus, da Erfahrungen des Menschen, die hier modelliert werden sollen, kein Ablaufdatum kennen.

Unterformen des XCS selbst basieren beispielsweise auf realen Werten statt binären (XCSR) [131], was bei der Verarbeitung reellwertiger Eingangsgrößen gewisse Vorzüge bringt. Der wesentliche Unterschied ist dabei, dass durch die Generalisierung der Wertebereich erweitert wird statt Joker an verschiedene Bitpositionen einzusetzen. In der vorliegenden Arbeit liegen sowohl reellwertige als auch logische Eingangsparameter vor, wie später gezeigt werden wird.

Außerdem wurden andere maschinelle Lernmethoden in LCS integriert, so in [45] eine Fuzzy-Lernkomponente, um die Exploration der Regeln gezielter durchführen zu können. Hierbei wird die Intelligente Explorationsmethode (IEM) vorgestellt, womit das XCSI entsteht. Die erreichten (leichten) Verbesserungen gegenüber dem klassischen XCS-Ansatz sind dort am Bei-

spiel eines Labyrinths gezeigt, in dem ein Agent so schnell wie möglich an sein Ziel kommen soll. Ein weiterer interessanter Ansatz ist die Integration des Backpropagationalgorithmus in LCS, der in der vorliegenden Arbeit nicht weiter verfolgt werden soll, da der Bezug zur zu modellierenden Umwelt nicht gegeben ist.

Nicht-markovsche und antizipative Ansätze

LCS gehen davon aus, dass die Umwelt als Markovmodell erster Ordnung interpretiert werden kann, bei der die aktuelle Situation nur von der direkt vorhergehenden Situation abhängt. Das ist oftmals nicht per se der Fall, da die Historie aus Vorzuständen eine Rolle bei der Wahl der optimalen Strategie in der aktuellen Situation spielt (Nicht-Markovsches Problem) [70]. Als Ausweg gibt es die Möglichkeit, alle relevanten Vorzustände in den aktuellen Zustand zu integrieren. Damit ist die Entscheidung bisher von einem einzigen Vorzustand, der alle nötigen Informationen über die Vergangenheit enthält, abhängig.

Die Aufspaltung des Nichtmarkovproblems in mehrere Markovsche als Alternative wird in [135] und [85], eigenen sich aber nicht als Methoden in der vorliegenden Arbeit. Diese Ansätze werden hier nur der Vollständigkeit halber genannt.

Des Weiteren werden bei XCSM beispielsweise interne Zustände in die Wissensbasis integriert, die die Vergangenheit abbilden [137] [71]. Diese Zusammenfassung besitzt entweder eine feste Anzahl historischer Zustände [76] oder eine variable [82]. Je nach Umwelt ist die eine oder die andere Variante diejenige, die zu einem besseren Ergebnis führt.

Eine weitere Möglichkeit ist es, nach der mathematischen Kettenregel, Zustände zu „überbrücken“. Damit ist es möglich, die gegebene Historie abzubilden und gut für die zu treffende aktuelle Entscheidung zu nutzen. Die wichtigsten Vertreter sind hierbei die CXCS [127], das ZCCS [126] sowie die Anticipatory Classifier Systems (ACS) [119].

Bei ACS werden antizipative Verhaltensweisen des Menschen, erklärt durch die kognitiven Lerntheorien als Methode der Psychologie, in das Modell der LCS integriert [50] [19]. Sie gehen im Wesentlichen auf Stolzmann [117] [118] zurück und sind für die hier gegebene Aufgabenstellung interessant.

Nach dem antizipativen Ansatz wird das Lernen als Erkenntnis, welcher Zustand zu welchem führt, gesehen [125]. Mit Hilfe dieser wird ein Umweltmodell im lernenden Menschen geformt, das ohne Belohnung der Umwelt auskommt und auf die Theorie von James zurückgeht [60]. Zur Abbildung dieses Modells ist eine bestimmte Anzahl von Classifiern nötig, die zu einer Population zusammengefasst werden, wie es beim LCS auch üblich ist. Der Antizipative Lernprozess (ALP) ist vergleichbar mit anderen LCS-Ansätzen und ist in Abbildung 21 gezeigt. Zuerst führt dabei ganz links in der Abbildung die Situation S_{start} zu einer Antwort (R) mit einer Antizipation (Erwartung) deren Konsequenzen, also des neuen Zustands der Umwelt nach dem durch den Lernenden gebildeten Umweltmodell. Danach werden die Antizipationen E_{ant} der generierten Effekte verglichen mit den wirklich eingetretenen E_{real} , wobei die Differenz (**comparison**) aus beiden den Modellfehler beschreibt. Wenn dieser einen gewissen Wert übersteigt, werden die benutzten Antizipationen geschwächt, ansonsten bestärkt. Völlig falsche Annahmen oder Erwartungen führen zur völligen strukturellen Modellkorrektur (**differentiation**).

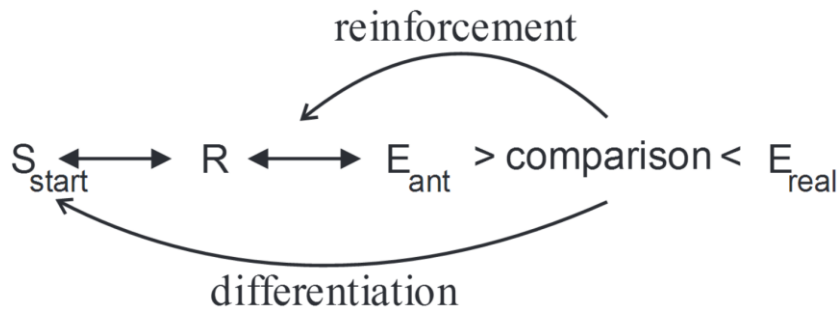


Abbildung 21. Der Antizipative Lernprozess (ALP) [19].

Praktisch werden die Regeln in ACS, um die Komponenten in Tabelle 3 erweitert, womit die Antizipation ausgedrückt wird. Das interne Umweltmodell wird durch die gemessene Genauigkeit q einer Regel präsentiert. Solange diese einen gewissen Schwellwert θ_i überschreitet, wird der durch die Regel beschriebene Zusammenhang ins Modell integriert.

Komponente	Formelzeichen Regel i	Beschreibung
Effekt	$\Psi_{i.E}$	Antizipierter Effekt bei Anwendung der Regel; Joker sind möglich
Markierung	$\Psi_{i.M}$	Zählt die nicht erfolgreiche Anwendung der Regel (Erfahrung der Antizipation)
Qualität	$\Psi_{i.Q}$	Gemessene Genauigkeit der Vorhersage als skalarer Wert
Nicht markiert	$\Psi_{i.u}$	Binärer Wert; ist „TRUE“, solange Antizipation stimmt

Tabelle 3. Nötige Erweiterungen der Regeln eines LCS, um ein ACS zu schaffen.

Interessant für die Modellbildung ist ebenfalls, dass ACS bei einem sehr allgemeinen Zustand beginnen. Das heißt, alle Regeln besitzen im Bedingungsteil $\Psi_i.s$ sowie im Effektteil $\Psi_i.E$ ausschließlich Joker. Damit ist zunächst kein Zusammenhang zwischen einer Situation und der zugehörigen Aktion hergestellt, das Modell ist maximal unscharf, unterliegt aber der Voraussetzung, dass Zusammenhänge zwischen Situationen und Aktionen existieren.

Nach Popper wird damit die Rationalitätshypothese [100] [129] angewandt, nach der Menschen situationsgerecht handeln (also nach individuellen Zielen strebend) [121]. Dabei haben Menschen, wie in ACS nachgebildet wird, zwar nie die vollständige Information zur Verfügung, treffen aber auf Grundlage der vorhandenen Informationen stets die optimale Entscheidung. Nach der bereits geschilderten Methode der abnehmenden Abstraktion (nach Lindenberg [77]) wird nun sukzessive das Modell der Umwelt ausgehend vom abstrakten Zustand gebildet. Dabei bildet das Gerüst des ACS selbst die Kerntheorie, auf deren Basis die Modellbildung erst möglich wird, und die Regeln sind die Brückenannahmen, die sich jederzeit (adaptiv) ändern können.

Ein weiterer regelbasierter Ansatz, antizipative Methoden in intelligente Lernnetzwerke zu integrieren, finden sich bei Drescher [29], wobei eine Regel aus den Komponenten: Kontext, Aktion und Ergebnis besteht. Dabei wird im Wesentlichen die Entwicklung von Kleinkindern als Grundlage benutzt. Auch die Dyna-Struktur von Sutton [122] basiert auf dem antizipativen Ansatz, bei dem ausschließlich globale, also generelle Umweltzusammenhänge dem Dyna-System präsentiert werden. Damit ist auch ohne die Anwendung eines Genetischen Algorithmus der Aufbau eines generalisierten Umweltmodells möglich.

Da die genetischen Algorithmen (GA) selbst lediglich für die Generalisierung der Wissensbasis sowie die möglichst maximale Gleichverteilung der Arbeitspunkte im Lösungsraum sorgen, wird das Lernen anhand präsentierter Lösungen als Antizipativer Lernprozess (ALP) zusammengefasst [132]. Dieser basiert auf dem Q-Learning und neigt zur Spezialisierung der Wissensbasis. Wie bei allen in LCS eingesetzten Lernverfahren wird auch hier online gelernt, während eine Generation von Classifiern einen Lösungsraum so gut als momentan möglich beschreibt.

Dabei werden die GA und der ALP parallel auf die Wissensbasis angewandt, wie in Abbildung 22 gezeigt ist. Wichtig dabei ist, dass die GA lediglich den Situationsteil einer Regel generalisieren, während der ALP Situationen und Effekte derselben spezialisiert (belehrt), womit der Evolutionsdruck aufrechterhalten wird.

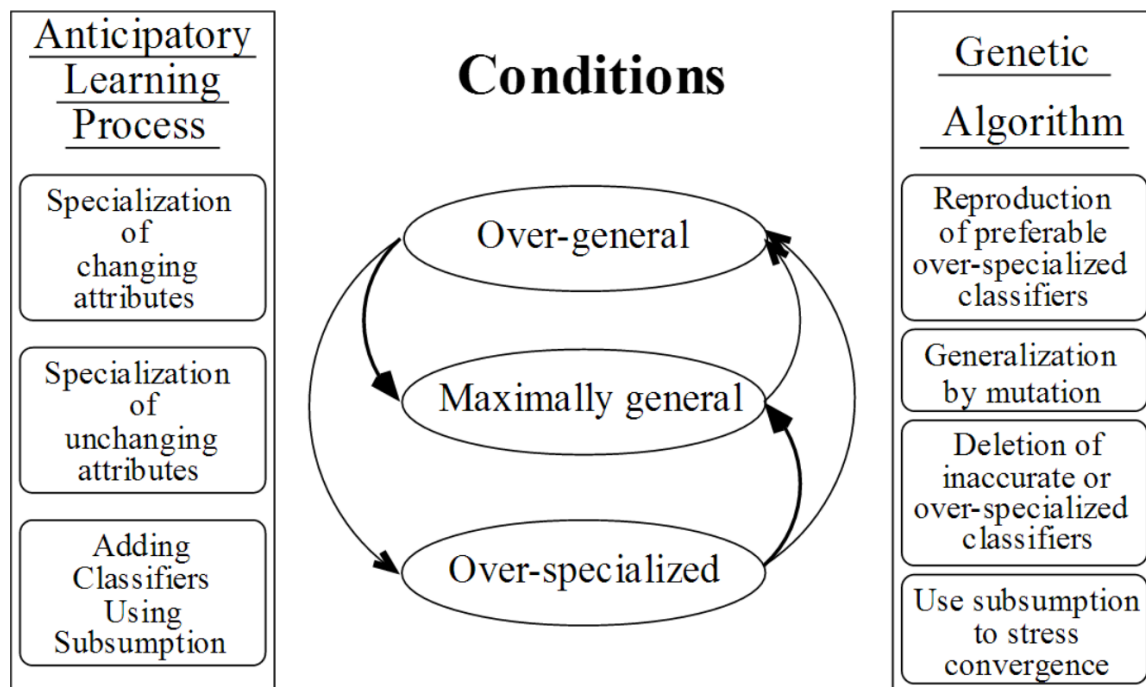


Abbildung 22. Das Zusammenspiel von GA und ALP [19].

ACS können mit den genannten Eigenschaften zur sehr groben Modellierung menschlichen Verhaltens herangezogen werden, die auf Antizipationen basieren. Dabei werden verhaltenstheoretische Erkenntnisse aus der Psychologie außer Acht gelassen. Deshalb ist eine Weiterentwicklung dieses Ansatzes für die bestehende Problemstellung nötig, nachdem gezielt die menschliche Entscheidungsfindung untersucht wurde. Weitere Modifikationen sind XCSF („XCS Function Approximator“, bei denen die Zielfunktion stückweise linear gelernt wird durch eine Gewichts Anpassung (durch Zerkleinerung des Lösungsraums) [138], und UCS, das den Ansprüchen des überwachten Lernens gerecht wird [11].

2.3.5 Die Building Blocks-Hypothese

Die wesentliche Stärke der Generalisierung durch den Einsatz von Jokern wird durch die Building Block-Hypothese, nach der ein genetischer Algorithmus die optimale Lösung anhand der Aneinanderreihung kurzer, aber leistungsstarker Schemata niedriger Ordnung (den Building Blocks) findet [49]. Diese Hypothese kommt bei jeder Regel in der Population [**P**] zum Tragen. Um das zu vereinheitlichen, wird deshalb zuerst näher auf das Schema-Theorem von Holland selbst eingegangen [51].

Für die Anwendung in einem Smart Home bedeutet das, dass beim Einschalten bestimmter Geräte nicht unbedingt jeder Parameter aus der Umwelt Berücksichtigung finden muss, sondern auch Muster erkannt werden können. Wenn beispielsweise das Licht in einem Zimmer immer zu einem bestimmten Tageszustand (wenn es dunkel wird) eingeschaltet wird, muss das nicht für jedes Zimmer separat gelernt werden, sondern dieses Erkenntnis aus dem ersten Zimmer kann auf andere übertragen werden.

Aufgrund der einfacheren Darstellung wird nachfolgend für den Einsatz in $\Psi_i.a$ und s_t von einem binären, kanonischen Alphabet Ω ausgegangen, das in (26) definiert ist, mit Hilfe dessen die aktuell gelesene Situation s_t gebildet wird. Bei einer Länge von λ Zeichen sind Ω^λ Situationen möglich.

$$s_t = \Omega^\lambda = \{0, 1\}^\lambda \quad (26)$$

Mit der Erweiterung des Alphabets um den Joker ergibt sich auf der anderen Seite für jeden möglichen Bedingungsteil einer Regel Ψ_i eine von $(\Omega+1)^\lambda$ Kombination, wie in (27) gezeigt ist [49]. Da sich die Regeln permanent ändern, wird hier ein bestimmter Zeitpunkt τ betrachtet.

$$\Psi_i.s(\tau) = (\Omega+1)^\lambda = \{0, 1, \#\}^\lambda \quad (27)$$

Ein Schema ist dabei ein String, der eine gewisse Anzahl von Jokern enthält. Damit stellt das es einen Unterraum des Lösungsraums dar (Abbildung einer bestimmte Menge von Lösungen). Das bedeutet, dass eine Eingangssituation s_t zu keiner, einer oder mehreren Regeln passt. In Abbildung 23 ist dabei ein Schema als dreidimensionale Lösung in einem vierdimensionalen Lösungsraum beschrieben, wobei der blaue, dreidimensionale Einheitswürfel dabei das Schema „1###“ im 4 D-Raum darstellt. Bildlich gesprochen beschreibt „jedes Schema eine Hyperebene im Hypereinheitswürfel (aber nur Ebenen, die parallel oder senkrecht zu Achsen stehen)“ [68].

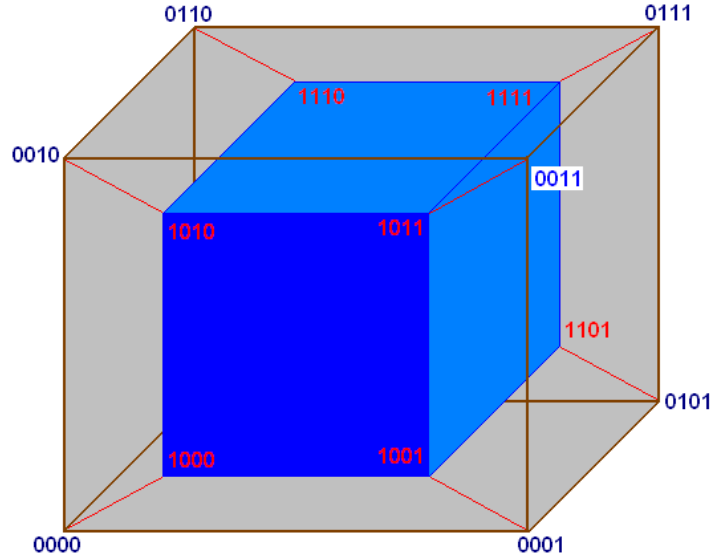


Abbildung 23. Ein vierdimensionaler Einheitswürfel (vierte Dimension durch unterschiedliche Farbgebung dargestellt) wird als Kombination zweier dreidimensionaler Einheitswürfel dargestellt. Jede Unterebene des vierdimensionalen Würfels ist eine gültige Lösung des vorliegenden Problems [eigene Darstellung].

Dabei ist das Schema nach Suchanek ein „[...] hochabstraktes[s] und sehr allgemeines[s] Grundmuster [...] [zur Problemformulierung und Analyse]“ [121]. Zusammenfassend ausgedrückt lassen sich komplexe Situationen einer Regel durch Schemata generalisieren [18], wobei auch die Positionen der Joker selbst eine Information über die Generalisierungsfähigkeit beinhalten.

Die Spezifität (oder Ordnung) $\text{ord}(\Psi_i)$ ist das Maß für den Abstraktionsgrad des Schemas einer Regel i und entspricht der Differenz aus der Länge $\text{len}(\Psi_i.s)$ des Situationsteils einer Regel i und der Anzahl der darin enthaltenden Joker ($= 1_{\{\#\}}$) und ist in (28) definiert [17] [38].

$$\text{ord}(\Psi_i) = \sum_{i=1}^{\text{len}(\Psi_i.s)} 1_{\{0,1\}}(\Psi_i.s) = \text{len}(\Psi_i.s) - 1_{\{\#\}} \quad (28)$$

Je kleiner die Spezifität einer Regel ist (also je mehr Joker vorhanden sind), desto abstrakter oder genereller ist die Regel. Da sowohl die Aktion als auch der zu erwartende Gewinn der Regel keine Joker besitzen, fließen sie nicht in die Definition ein.

Ein Extremfall eines Schemas ist die Ordnung $\text{ord}(\Psi_i) = 0$, bei der der abgebildete Unterraum ohne Joker gleichzeitig der gesamte Lösungsraum ist (wobei eine Interpretation als Schema hier grenzwertig ist) und der andere Extremfall ist $\text{ord}(\Psi_i) = \text{len}(\Psi_i)$, was einem singulären Arbeitspunkt im Lösungsraum entspricht.

Ordnung Ψ_i	Lampe	Radio
0 <i>(generellste Lösung)</i>	#	#
1	#	0
	#	1
	0	#
	1	#
2 <i>(speziellste Lösungen)</i>	0	0
	1	1
	0	0
	1	1

Tabelle 4. Die Ordnung eines Zustandsvektors mit der Länge 2.

Damit existieren im Lösungsraum 3^λ Schemata, wobei jede singuläre, individuelle Lösung wiederum Teil von 2^λ Unterräumen (Hypothesen) sein kann. Da wiederum bei jedem Schema der Joker an beliebigen Positionen i

($i = 0 \dots \lambda$) sitzen können, ergeben sich $\binom{l}{i}$ Positionen für die Joker. Das bedeutet wiederum, dass jedes Chromosom zu 2^l Schemata passt (s. Abbildung 24). Da wiederum $\|P\|^3$ Schemata in einer Population vorhanden sind, werden wesentlich mehr Schemata durchsucht als vorhandene Zeichen [140]. Dieses Phänomen wird als impliziter Parallelismus bezeichnet [49].

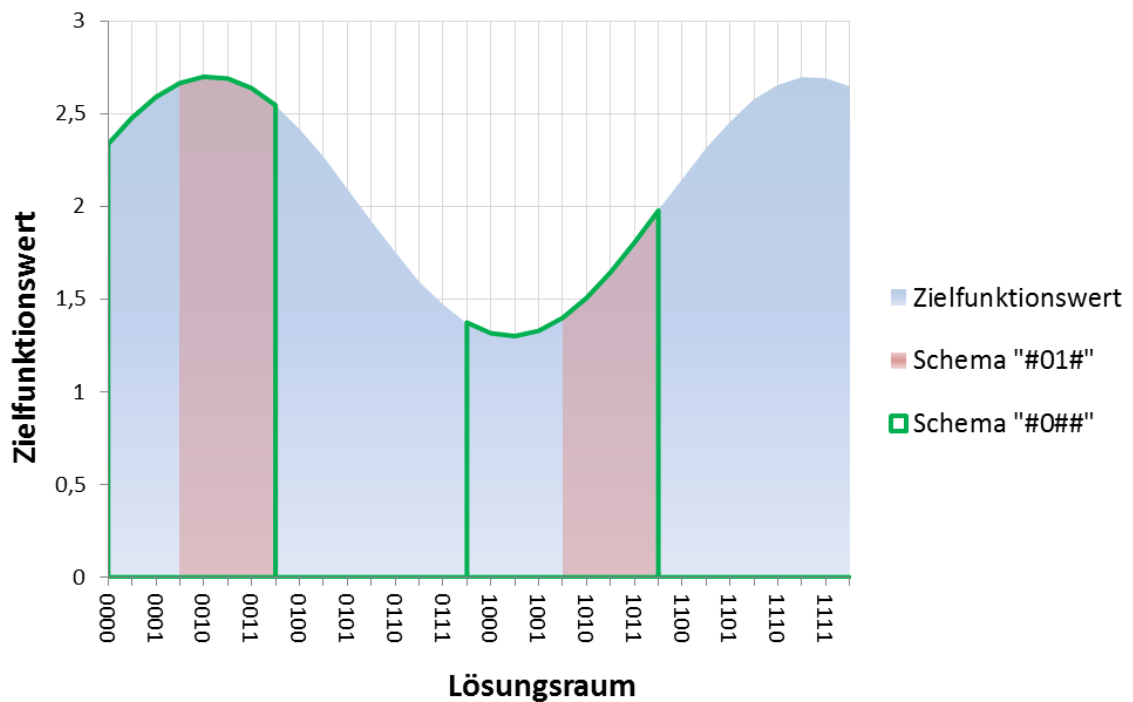


Abbildung 24. Grafische Darstellung von zwei Schemata. Der grün umrandete Bereich umfasst dabei das Schema „#0##“, wenngleich der rosa ausgefüllte Bereich das Schema „#01#“ beschreibt. Der blaue Bereich ist die kontinuierlich dargestellte komplette Lösungsmenge.

Um die Effizienz des Schematheorems bei der Suche nach der bestmöglichen Lösung zu verstehen, wird nun für die Vergleichbarkeit zweier aufeinander folgender Generationen (Zyklen) zu den Zeitpunkten t und $t+1$ ein geeignetes Maß für die Anpasstheit der Wissensbasis an die Umwelt gesucht. Dazu wird die mittlere Anzahl der zu $\Psi_{i,s}$ passenden statistisch verteilten Eingangssituationen s_t zu einem Zeitpunkt t (einer Generation) als

$N(\Psi_i.s, t)$ bezeichnet. Die Passung selbst ist in (29) definiert und wird durch das Formelzeichen „ \blacktriangleleft “ ausgedrückt („ $s_t \blacktriangleleft \Psi_i.s$ “ bedeutet, dass die gelesene Umweltsituation s_t zum Situationsteil der Regel i passt).

$$\sum_{\substack{i=0 \\ H_i \neq \#}}^{\lambda} \|s_t - \Psi_i.s\| = 0 \quad s_t \blacktriangleleft \Psi_i.s \quad (29)$$

Damit wird die Angepasstheit (Fitness) $f_{\text{abs}}(t)$ einer Eingangssituation s_t bezüglich der Population $\Psi_i.s$ zum Zeitpunkt t , kurz $s_t \blacktriangleleft \Psi_i.s$, definiert (s. (30)), wobei s_t für die standardisierte Eingangssituation repräsentativ für eine statistisch verteilte Anzahl von s_t zu einem beliebigen Zeitpunkt steht, der mit t bezeichnet wird. Von einer großen Angepasstheit von $\Psi_i.s$ spricht man dabei, wenn die gelesene Eingangssituation s_t durch viele Regeln $\Psi_i.s$ in der Population abgebildet werden kann.

$$f_{\text{abs}}(\Psi_i.s(t)) = \sum_{\substack{s_t \blacktriangleleft \Psi_i.s \\ \Psi_i.s \in \Psi_i.s(t)}} (\Psi_i.r) \quad (30)$$

Um die zeitliche Entwicklung der Wissensbasis bewerten zu können, wird die mittlere relative Fitness $f_{\text{rel}}(\Psi_i.s(t))$ normiert als die mittlere Angepasstheit der gelesenen Eingangssituation s_t bezüglich $\Psi_i.s$ zum Zeitpunkt t und ist in (31) dargestellt, wobei $N(\Psi_i.s, t)$ die mittlere Anzahl der Regeln in der Population zum Zeitpunkt t ist, die zur Eingangssituation s_t passen. Im Gegensatz dazu wird $f_{\text{rel}}(t)$ definiert als durchschnittliche Angepasstheit aller Regeln der Population zur aktuellen Situation s_t zum Zeitpunkt t .

$$f_{\text{rel}}(\Psi_i.s(t)) = \frac{f_{\text{rel}}(\Psi_i.s(t))}{N(\Psi_i.s, t)} \quad (31)$$

Die definierende Länge eines Schemas ($= dl(\Psi_i)$) wird in (32) definiert und ist identisch mit dem definierenden Anteil des Situationsteils einer Regel. Interpretiert werden kann es als Unterraum des Lösungsraums.

$$dl(\Psi_i) = dl(\Psi_i.s) = \max_{j \in \{1, \dots, \text{len}(\Psi_i.s)\}} \{h_j \neq "\#\"} - \min_{j \in \{1, \dots, \text{len}(\Psi_i.s)\}} \{h_j \neq "\#\"} \quad (32)$$

Ein String mit der Länge von 8 Zeichen und einer definierenden Länge von 5 ist als Beispiel in Abbildung 25 gezeigt. Die Joker in der Mitte des Strings spielen für die Berechnung keine Rolle.



Abbildung 25. Beispiel eines Strings mit der definierenden Länge von 5 (Länge: 8).

Die Suche nach der besten Lösung mit Hilfe der Building Blocks-Hypothese

Nach der Building Blocks-Hypothese wird die bestmögliche Lösung für ein gegebenes (komplexes) Problem durch die Bildung von kleinen, optimalen Teillösungen in Form von Hyperebenen (Schemata) im Lösungsraum repräsentiert¹³. Der entscheidende und nicht zu unterschätzende Vorteil der Anwendung genetischer Algorithmen gegenüber analytischen Lösungsansätzen

¹³ Deshalb wird die Building Block-Hypothese auch als Schematheorem bezeichnet.

ist dabei, dass infolge von Selektion, Rekombination und Mutation zielgerichtet der Anteil und die Güte der Teillösungen exponentiell steigen, womit eine sehr schnelle Verbesserung der gefundenen Gesamtlösung erreicht wird.

Auch die Lösungsfindung durch das menschliche Denken in unserem Alltag ist dadurch gut erklärbar. Stehen wir beispielsweise vor der Frage, in einem Haushalt das Radio oder den Fernseher für die neusten Nachrichten einzuschalten, kommt es zunächst darauf an, ob wir uns ausschließlich dem Medium widmen wollen (Fernseher) oder es nebenbei hören wollen und noch andere Aufgaben im Haushalt zu erledigen haben (Radio). Durch diese völlig unterschiedlichen Präferenzen werden auch für das gleichartige Problem, die Nachrichten zu erfassen, völlig andere Lösungen gewählt. Außerdem wird infolge der Information, dass eine für den Anwender sehr interessante Nachricht gesendet wird, eine viel speziellere Lösung angestrebt (tendenziell Fernseher).

Während des Entscheidungsprozesses sind Präferenzverschiebungen sowie das Hinzukommen oder Wegfallen von Einflüssen keinesfalls ausgeschlossen. In der vorliegenden Arbeit reichen bereits wenige Entscheidungen der Nutzer, Geräte in Abhängigkeit von Umweltbedingungen zu bedienen, um auf generelle Gesetzmäßigkeiten der Entscheidungsfindung des Anwenders zu schließen. Das trotzdem ein Entschluss des Modells mit großer Güte möglich ist, ist durch die Building Blocks erklärbar.

Um diese Hypothese zu erklären, wird die Verbesserung der relativen Angepasstheit (Fitness) $f_{\text{rel}}(\cdot)$ einer Generation der Population $[\mathbf{P}]_t$ zur nächsten $[\mathbf{P}]_{t+1}$ beschrieben. Zur Vergleichbarkeit wird s_t als standardisierte, normierte Regel konstant gehalten, was einer stabilen Umwelt entspricht. Alternativ könnte man auch die Wissensbasis mit allen jemals aufgetretenen Situatio-

nen beaufschlagen und daraus die mittlere Angepasstheit ermitteln. Da dadurch keine weiteren Informationen gewonnen werden (Referenzsituation kodiert ja alle anderen jemals aufgetretenen), wird von diesem deutlich aufwändigeren Weg abgesehen.

2.3.5.1 Einfluss der Selektion

Zuerst soll der Einfluss der Selektion (wird im Schritt 8 von Abbildung 17 im Rahmen der Genetischen Algorithmen durchgeführt) auf die relative Angepasstheit der Regelbasis (Population) $f_{\text{rel}}(\cdot)$ an die Referenzsituation s_t untersucht werden¹⁴. Ausgangspunkt der Überlegung ist, dass durch die Selektion per definitionem nur angepasste Regeln für die nächste Generation der Wissensbasis ausgesucht werden [41].

Die Wahrscheinlichkeit, dass eine Regel i (aus der Gesamtheit aller Regeln der Generation t) für die Selektion ausgewählt wird, wird dabei durch $p_i(s_t)$ ($0 \leq p_i(s_t) \leq 1$) ausgedrückt (s. (33)). Für die Herleitung wird das in der Praxis übliche Roulettewahlverfahren benutzt, nach dem jede Regel mit einer Wahrscheinlichkeit entsprechend ihrer Stärke (im konkreten Fall ist das die zu erwartende Belohnung der Regel $\Psi_{i,r}$) aus der Menge aller Regeln ausgewählt wird [145]. Diese Methode ist ein sehr guter Kompromiss im sogenannten „Exploration-Exploitation-Dilemma“, bei dem ein Ausgleich zwischen der Erforschung unbekannter Bereiche im Zustandsraum (Exploration) und der Verdichtung vorhandenen Wissens (Exploitation) angestrebt wird [35].

¹⁴ Die Definition ist in (31) gezeigt.

$$p_i(s_t) = \frac{\Psi_i \cdot r}{\sum_{j=0}^{\|P\|} \Psi_j \cdot r} \quad (33)$$

Es beschreibt $p_{\text{mean}}(s_t)$ die mittlere Überlebenswahrscheinlichkeit einer Regel, die zu s_t passt ($\Psi_i \cdot s \blacktriangleleft s_t$) und in (34) mit Hilfe von $p_i(s_t)$ berechnet wird. Der skalare Wert $N(s_t, t)$ misst die Anzahl der Regeln, die in der Generation t dabei zur Eingangssituation s_t passen.

$$p_{\text{mean}}(s_t) = \frac{\sum_{j=0}^{\|P\|} p_j(s_t)}{N(s_t, t)} \quad (34)$$

Außerdem wird die mittlere Stärke bzw. zu erwartende Umweltbelohnung der Regeln einer Generation, zu denen s_t passt, mit $r_{\text{mean}}(s_t)$ bezeichnet und anhand (35) berechnet.

$$r_{\text{mean}}(s_t) = \frac{\sum_{i=0}^{\|P\|} \Psi_i \cdot r(t)}{N(s_t, t)} \quad (35)$$

Im oben skizzierten Beispiel des Smart Home-Systems bedeutet das, dass zuerst im Denkprozess nur gültige Lösungen (Ein-/ Ausschalten von Geräten unter bestimmten Bedingungen), die den derzeitigen Bedürfnissen ent-

sprechen, für den weiteren Denkprozess zugelassen werden. Bei jeder „gedachten Lösung“, die nach der Auswahl noch übrig bleibt, ist eine durchschnittliche Umweltantwort (also die eigene Zufriedenheit) von $r_{\text{mean}}(s_t)$ zu erwarten.

$$\begin{aligned}
 p_{\text{mean}}(s_t) \cdot \|\mathbf{P}\| &= \frac{\sum_{j=0}^{\|\mathbf{P}\|} p_j(s_t)}{N(s_t, t)} \cdot \|\mathbf{P}\| = \frac{\sum_{j=0}^{\|\mathbf{P}\|} \Psi_j \cdot r}{\sum_{j=0_t}^{\|\mathbf{P}\|} \Psi_j \cdot r} \cdot \frac{\|\mathbf{P}\|}{N(s_t, t)} \\
 &= \frac{\frac{\sum_{j=0}^{\|\mathbf{P}\|} \Psi_j \cdot r}{\sum_{j=0_t}^{\|\mathbf{P}\|} \Psi_j \cdot r}}{\frac{N(s_t, t)}{\|\mathbf{P}\|}} = \frac{r_{\text{mean}}(s_t)}{r_{\text{mean}}} \quad (36)
 \end{aligned}$$

Wird jetzt die mittlere Überlebenswahrscheinlichkeit einer Regel aus der aktuellen Population $p_{\text{mean}}(s_t)$ mit der Anzahl der Regeln in der Wissensbasis¹⁵, $\|\mathbf{P}\|$, multipliziert, ergibt sich mit der Herleitung in (36) das Verhältnis aus der Angepasstheit der zu s_t passenden Regeln zur Angepasstheit aller.

Dabei spiegelt r_{mean} , wie bereits aus (36) ersichtlich ist, die mittlere zu erwartende Belohnung aller Regeln aus der aktuellen Wissensbasis wieder.

¹⁵ Während der Einlernphase variiert die Anzahl der Regeln in der Wissensbasis, wobei hier vom eingelernten Zustand ausgegangen wird, bei dem die maximal mögliche Anzahl von Regeln die Wissensbasis füllt.

Mit der Überlegung, dass per Definition die Selektion alle in der aktuellen Population vorhandenen Regeln, die zu s_t passen, auswählt und damit deren mittlere Überlebenswahrscheinlichkeit auf die gesamte Wissensbasis überträgt, ergibt sich im nächsten Schritt die mittlere Anzahl der zu s_t passenden Zeichen nach der Selektion $N(t+\Delta t_{\text{sel}})$ nach (37) (nach Einsatz von (34)).

$$N(s_t, t + \Delta t_{\text{sel}}) = \left(\sum_{\substack{j=0 \\ \Psi_j \cdot s < s_t}}^{\|P\|} p(\Psi_j) \right) \cdot \|P\| = N(s_t, t) \cdot p_{\text{mean}}(s_t) \cdot \|P\| \quad (37)$$

2.3.5.2 Einfluss der Rekombination

Im nächsten Schritt wird die mittlere Anzahl der zu s_t passenden Zeichen nach der Rekombination¹⁶ $N(t+\Delta t_{\text{sel}}+\Delta t_{\text{cross}})$ in (38) berechnet. Ausgehend von (37) werden nun die Regeln mit der Wahrscheinlichkeit p_{cross} durch die Rekombination verändert. Danach passen p_{loss} nicht mehr zu s_t und n_{gaines} Regeln passen erstmals zu s_t .

$$N(s_t, t + \Delta t_{\text{sel}} + \Delta t_{\text{cross}}) = (1 - p_{\text{cross}})(N(s_t, t + \Delta t_{\text{sel}})) + p_{\text{cross}}(N(s_t, t + \Delta t_{\text{sel}})(1 - p_{\text{loss}}) + n_{\text{gaines}}) \quad (38)$$

Die Wahrscheinlichkeit p_{loss} ist dabei maximal der Wahrscheinlichkeit, dass die Aufspaltung der Regeln ψ_{j1} und ψ_{j2} innerhalb der Nicht-Joker stattfindet multipliziert mit der Wahrscheinlichkeit, dass die zweite Regel ψ_{j2} nicht mehr zu s_t passt. Damit ergibt sich die Abschätzung in (39).

¹⁶ Hierbei handelt es sich um eine 1-Punkt-Rekombination (single-point crossover), bei der das Zielchromosom bis zu einem bestimmten Allel das eine Eltern-, ab diesem Allel das andere Elternchromosom enthält. Bei anderen Problemstellungen kann eine Mehr-Punkt-Rekombination (multi-point crossover) vorteilig sein [146].

$$p_{\text{loss}} \leq \frac{dl(s_t)}{\text{len}(s_t) - 1} \left(1 - \frac{N(s_t, t + \Delta t_{\text{sel}})}{\|P\|} \right) \quad (39)$$

Damit ergibt sich die Abschätzung von $N(s_t, t + \Delta t_{\text{sel}} + \Delta t_{\text{cross}})$ in Abhängigkeit der definierenden Länge $dl(\psi)$ der Regel in (40).

$$\begin{aligned} N(s_t, t + \Delta t_{\text{sel}} + \Delta t_{\text{cross}}) &\geq (1 - p_{\text{cross}}) (N(s_t, t + \Delta t_{\text{sel}}) \\ &+ p_{\text{cross}} \left[N(s_t, t + \Delta t_{\text{sel}}) \bullet \left(1 - \left(\frac{dl(s_t)}{\text{len}(s_t) - 1} \left(1 - \frac{N(s_t, t + \Delta t_{\text{sel}})}{\|P\|} \right) \right) \right) \right] \\ \stackrel{\text{ohne_gains}}{=} N(s_t, t + \Delta t_{\text{sel}}) &\left[1 - p_{\text{cross}} + p_{\text{cross}} \left(1 - \frac{dl(s_t)}{\text{len}(s_t) - 1} \left(\frac{N(s_t, t + \Delta t_{\text{sel}})}{\|P\|} \right) \right) \right] \quad (40) \\ &= N(s_t, t + \Delta t_{\text{sel}}) \left[1 - p_{\text{cross}} \frac{dl(s_t)}{\text{len}(s_t) - 1} \left(\frac{N(s_t, t + \Delta t_{\text{sel}})}{\|P\|} \right) \right] \\ &= N(s_t, t) \bullet p_{\text{mean}}(s_t) \bullet \|P\| \left[1 - p_{\text{cross}} \frac{dl(s_t)}{\text{len}(s_t) - 1} (1 - N(s_t, t) r_{\text{mean}}(s_t)) \right] \end{aligned}$$

Jetzt wird die mittlere Wahrscheinlichkeit, dass eine Regel der Wissensbasis zu s_t passt, als $P(s_t, t)$ definiert und durch die (41) berechnet.

$$P(s_t, t) = \frac{N(s_t, t)}{\|P\|} \quad (41)$$

Wenn in (40) der Zusammenhang aus (36) und (41) eingesetzt wird, ergibt sich eine weitere Vereinfachung in (42).

$$N(s_t, t + \Delta t_{sel} + \Delta t_{cross}) \geq P(s_t, t) \frac{r_{mean}(s_t)}{r_{mean}} \left[1 - p_{cross} \frac{dl(s_t)}{len(s_t) - 1} \left(1 - P(s_t, t) \frac{r_{mean}(s_t)}{r_{mean}} \right) \right] \quad (42)$$

2.3.5.3 Einfluss der Mutation

Bei der Mutation, die ebenfalls zu den Genetischen Algorithmen (Schritt 8 in Abbildung 17) gehört, werden einzelne Zeichen nach dem biologischen Vorbild in einem String zufällig verändert. Dazu sind mehrere Methoden denkbar. Alle sollen dazu führen, dass Bereiche des Zustandsraums erkundet werden, die bislang verborgen waren. Damit soll das „Hängenbleiben“ in lokalen Minima verhindert werden.

Die am häufigsten verwendete Methode ist die Punktmutation, bei der ein Zeichen zufällig innerhalb eines zulässigen Bereichs verändert wird. Diese Methode ist in Abbildung 26 dargestellt und ist interpretierbar als eine zufällige Veränderung in der Umwelt. Die Struktur des Zustandsraums bleibt bestehen.



Abbildung 26. An der zweiten Stelle im String findet eine Punktmutation statt.

Eine weitere Methode ist die Inversion. Hier wird ein Teilstring umgedreht [51], wie in Abbildung 27 gezeigt ist. Bei diesem Verfahren bleibt zwar die Dimension des Zustandsraums (Länge oder Anzahl der Bits des Zeichens) gleich, die Struktur jedoch ändert sich wesentlich. Denn wenn beispielsweise bestimmte Parameter in der Umwelt reellwertig und andere logisch verschlüsselt sind, können diese nicht vertauscht werden. Dann würde die Struktur der Umwelt nicht mehr korrekt abgebildet werden.

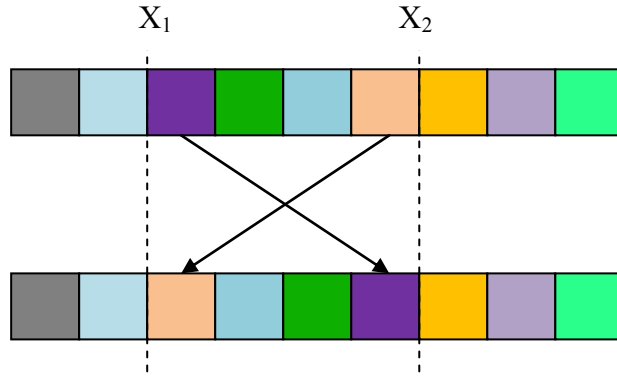


Abbildung 27. Inversion als Mutationsmethode.

Ein weiterer Ansatz ist die Deletion, bei der ein oder mehrere zufällige Zeichen aus dem String gelöscht werden. Außerdem können durch die Translation Teilstücke bzw. einzelne Zeichen im String an eine andere Stelle verschoben werden und durch Duplikation Teilstücke des Strings kopiert und an anderer Stelle eingefügt werden. Diese Verfahren ändern die Struktur der Umwelt wesentlich, und zwar in der Länge der Zeichen wie auch in der Syntax. Deshalb kommt diese in der vorliegenden Arbeit nicht zum Einsatz und wird nicht weiter verfolgt.

Deshalb konzentriert sich die weitere Abschätzung der mittleren Anzahl der Regeln, die zu s_t passen, auf die Punktmutation. Diese findet mit einer Wahrscheinlichkeit von p_{mut} statt. Die Abschätzung ist in (43) durchgeführt.

$$N(s_t, t + \Delta t_{sel} + \Delta t_{cross} + \Delta t_{mut}) \geq N(s_t, t + \Delta t_{sel} + \Delta t_{cross}) (1 - p_{mut})^{ord(s_t)} \quad (43)$$

Schlussendlich wird aus den vorangegangenen Gedankengängen das Schema-Theorem als mittlere Wahrscheinlichkeit, dass s_t zu den Regeln in **[P]** passt, in (44) abgeleitet.

$$P(s_t, t+1) \geq \frac{r_{\text{mean}}(s_t)}{r_{\text{mean}}} \left[1 - p_{\text{cross}} \frac{dl(s_t)}{len(s_t) - 1} \left(1 - P(s_t, t) \cdot \frac{r_{\text{mean}}(s_t)}{r_{\text{mean}}} \right) \right] \cdot (1 - p_{\text{mut}})^{\text{ord}(s_t)} P(s_t, t) \quad (44)$$

Aus (44) ergeben sich die Schlussfolgerungen, dass $P(s_t, t+1)$ umso größer wird (und damit die Wahrscheinlichkeit, dass die Eingangssituation s_t durch das Regelwerk der aktuellen Wissensbasis abgedeckt wird), je kürzer die definierende Länge $dl(s_t)$, je geringer die Ordnung $\text{ord}(s_t)$, und je größer die mittlere Bewertung $r_{\text{mean}}(s_t)$ ist. Diejenigen Hyperebenen, die diese Kriterien sehr gut erfüllen, vermehren sich von Generation zu Generation exponentiell und damit viel stärker als der Rest. Das bedeutet, dass diese den Lösungsraum besonders gut durchforsten. Sie werden Building Blocks genannt.

Deshalb werden beim Beispiel des Smart Home allgemeine Handlungsweisen auf verschiedene Umweltzustände übertragen, sodass es etwa keine Rolle spielt, ob es Sommer oder Winter ist oder in welchem Zimmer sich der Anwender befindet, wenn er bei Dunkelheit das Licht eingeschaltet haben möchte.

An dieser Stelle ist wichtig zu erwähnen, dass die Betrachtungen für unendlich große Populationen gelten. Umso kleiner $||[P]||$ ist, desto weniger Einfluss haben die Building Blocks auf die Lösungssuche. Desweiteren ist nicht sichergestellt, dass das gesuchte Schema überhaupt in $[P]$ enthalten ist neben der Tatsache, dass Schemata durchaus miteinander korrelieren können, was die Suche nach der optimalen Lösung deutlich erschwert.

Einige Ansätze sind bislang unternommen worden, um einige dieser Nachteile zu entschärfen. Der bekannteste ist die „Static Building Block Hypothesis“ von Grefenstette [42], der allerdings die Probleme nicht löst, sondern nur verlagert.

3 Grundlagen der menschlichen Entscheidungsfindung

Da die Problemstellung im Wesentlichen die Vorhersage von Entscheidungen im Haushalt beinhaltet, wird nachfolgend auf die menschliche Entscheidungsfindung näher eingegangen. Darunter versteht man, dass Menschen bewusst in einer bestimmten Situation σ_t zum Zeitpunkt t eine bestimmte Handlung α_t durchführen, die vorher unter Alternativen, die im Aktionsvektor $A(t)$ zusammengefasst sind (s. (45)), ausgewählt wurde.

$$A(t) = \begin{bmatrix} \alpha_1(t) \\ \alpha_2(t) \\ \dots \\ \alpha_m(t) \end{bmatrix} \quad (45)$$

Infolgedessen wird ein System (beispielsweise die Umwelt um den Entscheider herum) mit einer gewissen Wahrscheinlichkeit in einen neuen Zustand σ_{t+1} versetzt. Nachfolgend wird angenommen, dass diese Wahrscheinlichkeit 100 Prozent beträgt (Entscheidung unter Sicherheit), was einem vollständigen Durchgriff der gewünschten Zielsituation σ_{t+1}' und der tatsächlichen σ_{t+1} entspricht. Damit soll das Prinzip, das in der vorliegenden Arbeit später Anwendung findet, verdeutlicht werden.

Entscheiden ist der Prozess des Wählens zwischen mindestens zwei Alternativen mit dem Ziel, gewünschte Konsequenzen zu erreichen und unerwünschte Konsequenzen zu vermeiden. Der Prozess führt immer zu einer Entscheidung (Wahl). Durch die Entscheidung wird eine Option (Aktion) selektiert und als Realisierung eine Handlung ausgeführt, also ein Gerät ein- bzw. ausgeschaltet.

Definition 5. Entscheiden im Sinne der gegebenen Aufgabenstellung.

3.1 Normative Theorien

Die älteste Gruppe, die normativen Theorien, gehen von einem auf Beobachtungen basierenden Regelwerk als Grundlage für rationale menschliche Entscheidungen aus. Dieses Regelwerk wird empirisch gebildet und erforscht nicht die Gesetze, die diesen Regeln zugrunde liegen.

Damit eignet sich der normative Ansatz **nicht** als Basis für ENKOS, denn eine Vorhersage des Verbraucherverhaltens für unbekannte Umweltzustände/ Situationen eine Kernforderung bildet, die hiermit nicht erfüllt werden kann.

3.2 Präskriptive Modelle

Eine zweite Gruppe von Entscheidungstheorien basiert auf die präskriptiven Modelle, eine Art Erweiterung normativer Theorien. Diese Modelle beinhalten Regelwerke, in denen die typischen, aber realen Entscheidungssituationen (auf unterer Abstraktionsebene) beinhaltet sind.

Deshalb sind diese Modelle prinzipiell für eine Prognose des tatsächlichen Entscheidungsverhaltens von Menschen geeignet und im Sinne der vorliegenden Aufgabenstellung in ENKOS zu integrieren. Das Grundgerüst ist in

Prozess 1 gezeigt. Während theoretisch alle möglichen Konsequenzen in den ersten drei Schritten erfasst werden, stehen in realen Situationen nur eine gewisse Anzahl an Konsequenzen, die mit einem gewissen Zeitrahmen mit gewissem Aufwand erfasst werden können, zur Verfügung. Als Vereinfachung im vierten und fünften Schritt werden bei ENKOS alle Entscheidungen Niederschlag auf die Umwelt finden (Aktion a_{t1} führt unweigerlich zur neuen Situation s_{t1}). Der Expected Value (EV)-Wert ist dabei ein objektiver und quantifizierbarer Nutzen für den Anwender. Damit ist die Belohnung berechenbar, die dem Anwender zugutekommt, wenn er eine bestimmte Entscheidung trifft und ist damit ausschlaggebend für die Güte seiner Entscheidung.

- (1) Bestimme die Menge aller verfügbaren Optionen
- (2) Bestimme Konsequenzen aller Optionen
- (3) Bestimme den Wert jeder Konsequenz so genau und so objektiv wie möglich
- (4) Bestimmt die Wahrscheinlichkeit des Eintretens jeder Konsequenz so genau und objektiv wie möglich
- (5) Bilde für jede Konsequenz das Produkt aus Wert und Wahrscheinlichkeit
- (6) Bilde den EV für jede Option durch Summation der Wert-Wahrscheinlichkeits-Produkte
- (7) Entscheide Dich für die Option mit dem höchsten EV

Prozess 1. Entscheidungsfindung nach der deskriptiven Methode. Der vierte und fünfte Schritt sind Bestandteile des realen Prozesses, können jedoch durch eine Vereinfachung in ENKOS übersprungen werden und sind deshalb grau geschrieben. Der Schritt sechs hingegen vereinfacht sich, indem jeder EV lediglich dem zu erwarteten Wert entspricht.

Diese Schritte ähneln im Wesentlichen der Entscheidungsfindung in LCS. Während bei Präskriptiven Theorien im Schritt 7 ausschließlich exploitativ gearbeitet wird (Maximierungsprinzip), kommt im LCS die explorative Komponente durch Einsatz der Genetischen Algorithmen hinzu, indem sich mit einer kleinen Wahrscheinlichkeit sich auch für einen Schritt mit nicht dem größten EV entschieden wird.

Desweiteren kommen im LCS die Verarbeitung der Rückmeldung aus der Umwelt zur Evaluierung der getroffenen Entscheidung (sowie aller Regeln, nach deren Anwendung es zur gleichen Entscheidung geführt hätte) hinzu sowie die Weiterentwicklung des Entscheidungsmodells mit Hilfe von Erfahrungen (Exploration) neben Generalisierung (Exploitation). Damit ist das Resultat präskriptiver Entscheidungstheorien, im Gegensatz zu deskriptiven, ein normatives Modell.

3.3 Deskriptive (empirische) Theorien

Die deskriptiven Theorien bilden, im Gegensatz zur ersten Gruppen, die tatsächlichen, empirisch ermittelten Entscheidungen des Menschen ab. Im Rahmen dieser wird angestrebt, die Gesetzmäßigkeiten zu erkennen, die hinter real getroffenen Entscheidungen stecken. Damit entsteht als Ergebnis ein Modell des tatsächlichen Entscheidungsverhaltens von Menschen.

Zwei Thesen von Herbert A. Simon bilden die Grundlage für deskriptive Theorien [112]. Zum einen werden durch Menschen allgemein Entscheidungen getroffen, die gerade so gut sind, den Entscheider zufrieden zu stellen, und nicht etwa optimale Entscheidungen. Das wird als Satisfizierungskonzept oder Anspruchsniveaue Anpassung bezeichnet. Zum anderen sind – im Gegensatz zu präskriptiven Theorien – die Alternativen nicht a priori be-

kannt, sondern werden iterativ durch den Zugewinn an Erfahrungen ermittelt.

Ziel dieser Arbeit ist eine Vorhersage für unbekannte Arbeitspunkte im Umweltzustandsraum. Ein weiteres wesentliches Ziel ist die Interpretation der adaptiv aufgestellten Regeln, womit ein Prozessmodell (White Box-Ansatz) von der Entscheidung als Grundlage für ENKOS nötig wird. Im Gegensatz zum Strukturmodell (Black Box-Ansatz) werden hier die zugrunde liegenden Regeln, die für die Entscheidung benutzt werden, herangezogen.

Die Entscheidungsregel ist, wie bei den beiden vorher genannten Theorien, auch hier das Maximierungsprinzip. Der Unterschied dabei ist, dass diesmal nicht der EV bewertet wird, sondern der Nutzen für das Individuum (vgl. Schritte 6 und 7 in Prozess 1). Aus verschiedenen Definitionen (als Grundlagen dienen [13], [61], [98]) dieses Nutzens wird der Begriff in der vorliegenden Arbeit in Definition 6 definiert.

Der (antizipierte) **Nutzen** ist der zu erwartende, vorgestellte messbare Gewinn einer Entscheidung, sofern diese getroffen wird und der damit erzwungene neue Umweltzustand eingetreten ist.

Definition 6. (Antizipierter) Nutzen – vergleichbar mit dem „Reward“ einer Regel.

Während ein Zyklus eines LCS mit einer präskriptiven Entscheidung des Menschen vergleichbar ist, arbeitet dieses infolge der Anwendung Genetischer Algorithmen wie ein deskriptives Modell. Anders formuliert wird der Suchprozess dadurch verbessert, dass permanent unbekannte Arbeitsbereiche erschlossen werden, aber permanent optimale Entscheidungen bezüglich des aktuell vorhandenen Wissensstandes getroffen werden.

Um zu erklären, wie α_t aus allen Handlungsalternativen ausgewählt wird, wählt man die deskriptive Entscheidungstheorie, die wiederum aus der präskriptiven (normativen) entwickelt wurde. Während nach dieser erklärt wird, wie der Entscheider unter idealen, rationalen Bedingungen mit einem widerspruchsfreien System der Umwelt, einer unbegrenzten Rechenkapazität sowie aller nötigen Informationen die Alternative auswählen sollte, beschreibt die zweite den tatsächlichen, realen Entscheidungsprozess. Dabei werden eine limitierte Kapazität (sowohl intellektuell als auch zeitlich), eine nicht widerspruchsfreie Welt sowie das Vorhandensein nicht aller Informationen zum Zeitpunkt der Entscheidungen zugrunde gelegt. Diese Theorie wird nachfolgend weiterverfolgt, weil dem vorliegenden Problem eine Realität mit all seinen natürlichen Beschränkungen zugrunde liegt.

Im Allgemeinen wird von einer Modellbildung der Beobachtungen bzw. dieser Realität (Umwelt) durch den menschlichen Entscheider und der Wissensableitung durch Hypothesen und Sätze ausgegangen, um eine Handlung in einer gewissen Situation σ_t auszuwählen, die den zu erwartenden Nutzen maximiert (Maximierungsprinzip), wie in (46) zu sehen ist. Hier stellt $\Phi(\alpha^*)$ den Nutzen der besten Entscheidung aus dem Vektor $\Phi(\alpha)$ dar. Das erklärende Modell ist in Abbildung 28 gezeigt.

$$\Phi(\alpha^*) = \max_{\alpha \in A} \Phi(\alpha) = a_t \quad (46)$$

So werden im ersten Schritt in Abbildung 28 aus den Beobachtungen mittels Induktion (Generalisierung) die Hypothesen (Wissensbasis) erst aufgebaut, bestehende bestätigt oder widerlegt (Modellbildung und -verifikation), bevor im zweiten Schritt durch die Deduktion wiederum allgemeine Hypothesen auf die konkrete Situation angewandt werden (Modellanwendung).

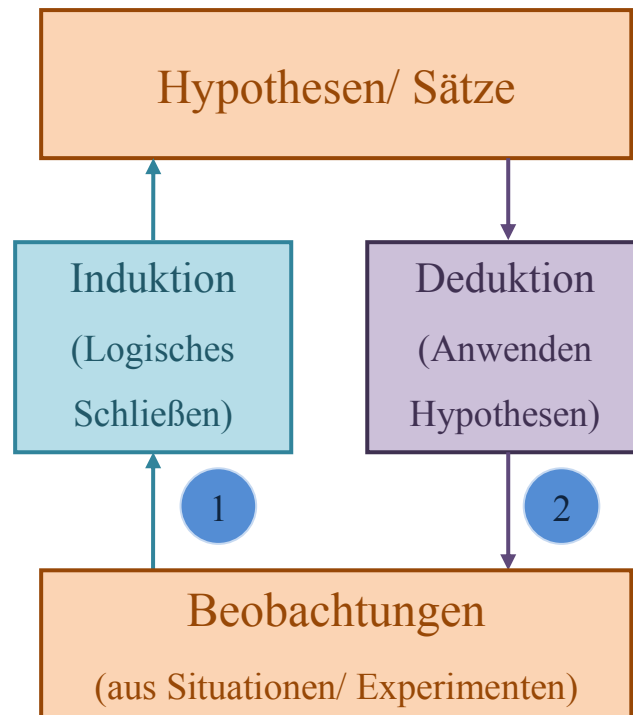


Abbildung 28. Entscheidungsprozess anhand der deskriptiven Theorie. Zuerst wird (linke Seite) per Induktion aus gegebenen Beobachtungen ein Hypothesenwerk (Wissensbasis) aufgebaut, die wiederum per Deduktion im nächsten Schritt in speziellen Situationen Anwendung finden.

Da sich ein Modell permanent in einem bestimmten Entwicklungsstadium befindet (je nach bislang bereits präsentierten Arbeitspunkten via Lernbeispiele), werden generell keine optimalen Entscheidungen durch Menschen getroffen [112], sondern lediglich momentan zufrieden stellende, was als Satisfizierungskonzept beschrieben wird¹⁷. Desweiteren ist ein Such- und Vergleichsprozess nötig, um aus allen bekannten Alternativen die beste auszuwählen. Prinzipiell ist dabei zu beachten, dass Kriterien je nach zu erreichendem Ziel anders gewichtet werden (keine Präferenzstabilität).

Um das zu verdeutlichen, wurden von Cyert und March [24] die wesentlichen Einflussfaktoren für eine Entscheidung beispielhaft am Führen eines

¹⁷ Leitmotiv ist hier: „So gut wie nötig“

Unternehmens definiert (für andere Anwendungsfälle müssen diese Faktoren angepasst werden):

- ➔ Beschränkung auf die wesentlichen Ziele,
- ➔ Vermeidung von Ungewissheit,
- ➔ Suche nach zieladäquaten Lösungen und
- ➔ Organisatorisches Lernen.

In der vorliegenden Arbeit sind die Kriterien für entsprechende Entscheidungen und deren Gewichtung für die Personen in der Wohnung ins Modell zu integrieren, was nicht durch jeden Modellansatz möglich ist. Desweiteren muss der Reifegrad des Modells Berücksichtigung finden, da im Gegensatz zur „klassischen“ Modellbildung Gewohnheiten sich permanent ändern und neu gelernt werden müssen. Desweiteren muss das Modell die Struktur der menschlichen Entscheidungsfindung abbilden.

Da der Nutzen selbst durch den Menschen interpretiert und je nach äußeren Umständen gewichtet wird, muss die objektive Messung (= Wert) einer zu erwartenden Belohnung oder Bestrafung erst noch subjektiviert werden, was durch die deskriptiven Entscheidungstheorien beschrieben wird. Da sich dieser Prozess innerhalb des menschlichen Denkens abspielt, ist das Teil des Denkmodells und muss sich auch im ENKOS wiederfinden.

Die Subjektivierung ist zwar von den Präferenzen und menschlichen Charakterzügen abhängig, trotzdem lassen sich zwei prinzipielle Regeln erkennen. Zum einen würde beispielsweise die Steigerung eines zu erhaltenden Gewinns von 1000 € um weitere 10 € weniger subjektiven Mehrnutzen in der menschlichen Empfindung erzeugen als die gleiche Steigerung bei einem Gewinn von ohnehin bereits 500 € (auf nun 510 €). Dieser allgemein beobachtbare Effekt wird als *dimishing sensitivity* bezeichnet und lässt sich

in der konkaven Kurvenform für Gewinne ($\text{Gewinn} > 0 \text{ €}$) und konvexen Form im Verlustbereich in Abbildung 29 ablesen.

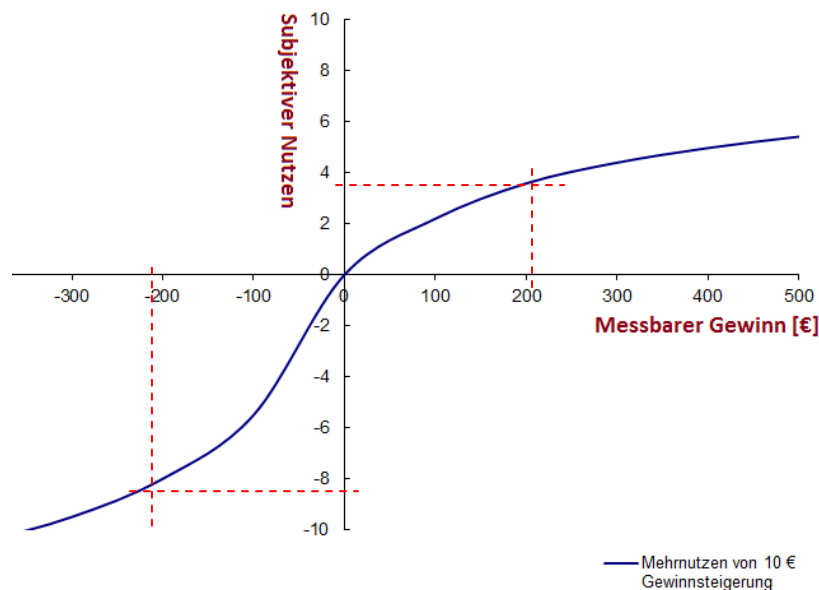


Abbildung 29. Das Nutzen-Gewinn-Diagramm beschreibt den subjektiven Nutzen in Abhängigkeit vom realisierten (objektiven) Gewinn für ein typisches Beispiel. Die Funktion ist durch eine logarithmische beschreibbar.

Zum anderen hat ein bestimmter Verlust stärkeren Einfluss auf den subjektiven Nutzen als ein betragsgleicher Gewinn, was durch die größere Steilheit der Funktion im Bereich $\text{Gewinn} < 0 \text{ €}$ ausgedrückt wird. Dieses Phänomen wird als **loss aversion** bezeichnet und drückt sich in Abbildung 29 darin aus, dass beim Gewinn von 200 € circa vier Einheiten subjektiver Nutzen empfunden werden (rote gestrichelte Linie), während beim Verlust von 200 € circa -8 Einheiten empfunden werden. Beide allgemein gültigen Beobachtungen fließen später in die Modellbildung ein.

4 Definition der Zielfunktion für ENKOS

4.1 Anforderungen an ENKOS und Abgrenzung zu bisherigen Systemen

Zwei zentrale Anforderungen werden an das Energie- und Komfortmanagementsystem (ENKOS) gestellt. Zum einen soll der Energieverbrauch in einem abgeschlossenen Haushalt minimiert und gleichzeitig der Wohnkomfort für alle Anwender maximiert werden. Die sich daraus ergebenden Teilziele werden in diesem Kapitel später erläutert und zu bestehenden Systemen hin abgegrenzt, um zum Schluss eine Zielfunktion als objektive Gütebewertung für ENKOS zu definieren. Diese soll den Ausgleich zwischen beiden Teilzielen realisieren.

Dabei können Situationen auftreten, bei denen die Handlungsalternativen einen Konflikt zwischen beiden Zielen ausgleichen müssen, wenn die Senkung des Energiebedarfs gleichzeitig zu Kosten des Komfort geht, aber auch sich verstärkende. Damit fließen, im Gegensatz zu den vorgestellten Systemen am Markt bzw. in der Literatur, neben messbarem Stromverbrauch auch menschliche Entscheidungstheorien in die Zielfunktion ein.

4.2 Grundstruktur der Zielfunktion

Die Zielfunktion f_{ENKOS} besteht aus der Suche der beiden Teilziele, nämlich der Energieminimierung (oder auch Maximierung der Energieeinsparung), sowie der Komfortmaximierung, aufintegriert über ein Jahr (s. (47)). Beide Summanden sollen so gewichtet werden, dass sie jeweils 50 Prozent des Funktionswerts von $f_{\text{ENKOS}}(t)$ realisieren. Der Wertebereich von f_{kont} in (47)

soll dabei auf 0...1000 normiert werden. Das bedeutet für die beiden Summanden $f_{\text{comfort}}(t)$ und $f_{\text{energy}}(t)$ einen Wertebereich von jeweils 0 bis 500.

$$f_{\text{ENKOS, kont}} = \int (f_{\text{comfort}}(t) + f_{\text{energy}}(t)) dt \quad (47)$$

Da im vorliegenden Fall eine zeitkontinuierliche Simulation deutlich die Kapazitäten der Rechentechnik überschreitet, wird als Vereinfachung nur jede halbe Stunde ein Ereignis zugelassen und der Zeitbereich zwischen zwei Ereignissen konstant gehalten (Sample and Hold-Verfahren). Damit wird der kontinuierliche Bereich für ein ganzes Jahr in 17520 äquidistante Halbstundenschritte (mit h' bezeichnet, vgl. (49)) zerlegt und f_{ENKOS} vereinfacht sich zu einer Summenfunktion über das gesamte Jahr in (48). An der Methodik der Definition ändert das nichts.

$$f_{\text{ENKOS}} = \sum_{t=0}^{17520h'} (f_{\text{comfort}}(t) + f_{\text{energy}}(t)) \quad (48)$$

Das Halbstundenintervall wurde gewählt, weil die meisten Handlungen im häuslichen Alltag deutlich länger als eine halbe Stunde andauern zumal eine Statistik über ein Jahr geführt wird, sodass hier eine angemessene Abtastrate vorhanden ist und gleichzeitig vertretbare Rechenzeiten realisiert werden können. Die Erfahrungswerte stammen aus längerfristigen Messungen mehrerer Haushalte in Deutschland, die von der Firma Aizo erhoben und anonymisiert zur Verfügung gestellt wurden [2].

$$1 h' = \frac{1}{2} h \quad (49)$$

Das beste Ergebnis des zu entwickelnden Systems ist erreicht, wenn f_{ENKOS} in (48) maximal wird. Damit ist das Optimalitätskriterium, ausgedrückt in (50), erreichbar durch die Maximierung der beiden Teilausdrücke. Der Zeitbereich von einem Jahr als Vergleichsgrundlage wurde gewählt, da hier alle saisonalen (z. B. Jahreszeiten) sowie täglichen Effekte enthalten sind. Sobald das beste strukturelle Modell für ein Jahr gefunden ist, werden auch Zeitintervalle über mehrere Jahre untersucht.

$$\text{Ziel: } \max(f_{\text{ENKOS}}) \quad (50)$$

Nachfolgend werden die Teilzielfunktionen definiert, um das Modell zu bilden, die laut Definition in (50) ebenfalls maximiert werden müssen.

4.3 Definition der Teilzielfunktion f_{comfort} für die Komfortmaximierung

Aus der Forderung in (50) ergibt sich für die Komfortmaximierung, dass sie ihren maximalen Wert annimmt, wenn der Anwender alle seine Wünsche erfüllt sieht. Das ist genau dann erreicht, wenn die Schaltung¹⁸ aller Geräte im Haushalt korrekt erfolgt, was zunächst eine zuverlässige und fehlerfreie Vorhersage des Verbraucherwunschs bedeutet. Um das messen zu können, wird die absolute Fehlsteuerung definiert.

¹⁸ Die „Schaltung“ eines Geräts ist entweder ein Einschalt- oder ein Ausschaltvorgang.

Die **absolute Fehlsteuerung** $S_{\text{fail}}(t)$ in einem Haushalt gibt an, wie viele Geräte zum Zeitpunkt t falsch geschaltet werden. Ein Fehlschaltvorgang liegt vor, wenn der Anwender ein Gerät deaktiviert haben möchte und ENKOS es aktiviert, und umgekehrt. Sind alle Geräte richtig angesteuert, beträgt $S_{\text{fail}}(t) = 0$, sind alle genau falsch aktiviert, ist er gleich der Anzahl der Geräte im Haushalt (nachfolgend als $\#Dev$ bezeichnet). $S_{\text{fail}}(t)$ kann für jeden einzelnen Zeitschritt berechnet werden und ist also zeitabhängig.

Definition 7. Absolute Fehlsteuerung $S_{\text{fail}}(t)$ als Maß für den Verbraucherkomfort.

Die **relative Fehlsteuerung** $s_{\text{fail}}(t)$ drückt die Anzahl der fehlgesteuerten Geräte (absolute Fehlsteuerung) pro Gesamtanzahl der Geräte ($\#Dev$) im betrachteten Haushalt aus (vgl. (51)). Damit kann $s_{\text{fail}}(t)$ Werte in den Grenzen 0 (alle Geräte richtig im Sinne des Anwenders aktiviert) und 1 (alle Geräte falsch aktiviert) annehmen.

Definition 8. Relative Fehlsteuerung $s_{\text{fail}}(t)$ als Maß für den normierten Verbraucherkomfort.

Damit ergibt sich als Optimierungsziel, die relative Fehlsteuerung zu minimieren, um die Zielfunktion zu maximieren. Deshalb wird eine weitere Definition nötig, die dieses Verhältnis umkehrt und die es zu maximieren gilt: die korrekte Ansteuerung $s_{\text{kor}}(t)$ definiert als relativer Wert über die korrekte Ansteuerung der Geräte.

$$s_{\text{fail}}(t) = \frac{S_{\text{fail}}(t)}{\#Dev} \quad (51)$$

Eine weitere Information steckt in der Zeit, bis ein Nutzer eine Fehlentscheidung von ENKOS korrigiert. Dafür wird der Faktor $k_{\text{Korrektur}}$ definiert,

dessen Werte in Tabelle 5 definiert sind. Die Ermittlung der Korrekturzeit wird realisiert, indem der Zielzustand, wenn möglich, langsam (innerhalb einer Minute) angesteuert wird. Bei einem Einschalten des Lichtes wird also eine Minute lang das Licht hochgedimmt, bis es vollends leuchtet. Ab Beginn dieser Übergangszeit hat der Mensch die Möglichkeit, den Zustand zu bewerten. Lässt er das Licht eingeschaltet, signalisiert er volle Zufriedenheit. Eine Korrektur also Ausschalten des entsprechenden Lichts, wird als Unzufriedenheit interpretiert. Die Zeitdauer bis zur Korrektur ist gleichzeitig ein Maß für diese Unzufriedenheit, denn ein sofortiges Eingreifen (abzüglich der Reaktionszeit) wird als sehr unzufrieden interpretiert, während eine späte Korrektur suggeriert, dass der Anwender selbst unsicher ist, da er den Zustand der eingeschalteten Lampe als inakzeptabel empfindet. Deshalb handelt sich in diesem Fall um eine recht geringe Unzufriedenheit.

Durchschnittliche Korrekturzeit durch den Anwender	$k_{\text{Korrektur}}$
0 s	1,0
0 s ... 60 s	1,0 ... 0,5 (linear)
60 s	0,5

Tabelle 5. Der Faktor $k_{\text{Korrektur}}$ abhängig von der Korrekturzeit (des Anwenders).

Im Gegensatz zur rein qualitativen Aussage, wie in [91] beschrieben ist, wird hier die Zufriedenheitsfunktion auf der Basis um die quantitative Auswertung der Korrekturzeit der einzelnen Geräte. Die Werte zur Berechnung der Parameter sind in Tabelle 5 zusammengestellt, woraus sich (52) ergibt.

$$s_{\text{kor}}(t) = [1 - s_{\text{f\ddot{a}il}}(t)] \bullet k_{\text{Korrektur}} = \left[1 - \frac{s_{\text{f\ddot{a}il}}(t)}{\#Dev} \right] \bullet k_{\text{Korrektur}} \quad (52)$$

Wie in Kapitel 3 bereits erl\ddot{a}utert wurde, spiegelt $s_{\text{kor}}(t)$ noch nicht die subjektive Nutzerzufriedenheit wider, sondern erst den objektiven, messbaren Gewinn. Denn die Erh\ddot{o}hung von $s_{\text{kor}}(t)$ um einen Wert Δs_{kor} verursacht bei zwei verschiedenen Ausgangswerten von $s_{\text{kor}}(t)$ auch zwei unterschiedliche Verbesserungen der Nutzerzufriedenheit (in Kapitel 3 als „Diminishing sensitivity“ erl\ddot{a}utert). Desweiteren hat eine \dd{A}nderung von $s_{\text{kor}}(t)$ um Δs_{kor} eine andere Zufriedenheitssteigerung als $s_{\text{kor}}(t)$ um $-\Delta s_{\text{kor}}$ einen Zufriedenheitsverlust nach sich ziehen w\dd{u}rde (Risikoaversion). Deshalb wird im n\dd{a}chsten Schritt eine Subjektivierung durchgef\dd{u}hrt.

Daf\dd{u}r wird als Methode der Entscheidungspsychologie die **Nutzentheorie** genutzt, die von Bernoulli erstmals formuliert wurde. Er geht davon aus, dass nicht objektive, sondern der subjektive Wert einer Konsequenz die Entscheidung bestimmt [12]. Dieser Wert wird als Nutzen f\dd{u}r den Entscheider bezeichnet. Die Funktion selbst wird in zwei Schritten definiert (analog Abschnitt 3.3):

- (1) Die grunds\dd{a}tzliche Form der Funktion ist als erstes zu bestimmen. Dabei sind eine logarithmische (risikoaverse), eine lineare (analog Wert-Erwartungs-Theorie) oder eine exponentielle (risikofreudige) Grundform m\dd{o}glich.
- (2) Durch experimentelle und/ oder analytische, psychische Nachweise werden im zweiten Schritt die Parameter f\dd{u}r die Funktion per Regressionsverfahren definiert.

Im vorliegenden Fall wird diese Methodik genutzt, um die Abbildung der subjektiven Zufriedenheit $f_{\text{comfort}}(t)$ zu jedem Zeitpunkt über die durch EN-KOS getroffene Entscheidung auszudrücken. Diese Zufriedenheit wird in Abhängigkeit der korrekt geschalteten Geräte definiert. Für die Ermittlung der prinzipiellen Kurvenform werden zwei Arbeitspunkte festgelegt:

Randbedingung 1: Wenn der Nutzer eine völlige Fehlansteuerung ($s_{\text{kor}}(t) = 0$) vorfindet und plötzlich ein Gerät richtig angesteuert wird ($0 < s_{\text{kor}}(t+1) \ll 1$), wird das seine Zufriedenheit nur unwesentlich steigern, also ist der Gradient an dieser Stelle sehr klein und kann zur Definition der Funktion mit 0 angenommen werden (s. (53)).

$$\left. \frac{df_{\text{comfort}}(t)}{dt} \right|_{s_{\text{kor}} \approx 0} = 0 \quad (53)$$

Randbedingung 2: Bei einer fast perfekten Ansteuerung aller Geräte jedoch ($0 \ll s_{\text{kor}}(t) < 1$) fällt dem Nutzer jeder noch so kleine Fehler sehr auf, deshalb ist der Gradient hier viel größer als im ersten Fall (s. (54)) und der Funktionswert erreicht $f_{\text{comfort,max}}$ (den maximal möglichen Wert im Wertebereich der Funktion $f_{\text{comfort}}(t)$, was in (55) verarbeitet wird).

$$\left. \frac{df_{\text{comfort}}(t)}{dt} \right|_{s_{\text{kor}} \approx 1} \gg 0 \quad (54)$$

$$f_{\text{comfort}}(t)|_{s_{\text{kor}}(t)=1} = f_{\text{comfort,max}} \quad (55)$$

Damit ergibt sich prinzipiell eine konvexe (oder exponentielle) Funktion für $f_{\text{comfort}}(s_{\text{kor}})$ mit den Randbedingungen, dass bei völliger Fehlansteuerung auch die Zufriedenheit absolut nicht gegeben ist. Die Struktur dieser Funktion mit den Parametern a , b und c , die den Randbedingungen gerecht wird, ist in (56) dargestellt.

$$f_{\text{comfort}}(s_{\text{kor}}) = a e^{b s_{\text{kor}}} + c \quad (56)$$

Aus der Randbedingung 1 ergibt sich $c = -1$ sowie $a \neq 0$, um in jedem Fall die Kurve durch den Nullpunkt des Diagramms zu führen. Für die beiden anderen Werte gibt es verschiedene Parameterpaare, die die zweite Randbedingung erfüllen würden. Um diese zu bestimmen, werden in der Literatur mehrere Varianten vorgeschlagen, deren Modifizierung auf die Parameterbestimmung der Zufriedenheitsfunktion geprüft wird.

- (1) Bei der **Halbierungsmethode** wird zunächst die halbe subjektive Zufriedenheit als Stützstelle von $s_{\text{kor}}(t)$ in Abhängigkeit der relativen korrekten Ansteuerung zwischen den äußeren Randpunkten $s_{\text{kor}}(t) = 0$ und $s_{\text{kor}}(t) = 1$ bestimmt. Danach werden beide Intervalle wiederum halbiert bezüglich Zufriedenheit und die jeweiligen Mittelpunkte bestimmt. Dieses Verfahren kann dann rekursiv für die jeweils entstehenden Stützstellen fortgesetzt werden. Zum Schluss wird die Funktion an alle Stützstellen approximiert (oder bei nur einer Stützstelle

analytisch ermittelt). Diese Möglichkeit ist gut auf die Parameterdefinition der Zufriedenheitsfunktion übertragbar.

- (2) Die **Verteilungsmethode** werden Punkte auf verschiedene Stützstellen (Unterziele), ausgehend vom Zielzustand ($s_{\text{kor}}(t) = 1$) verteilt. Diese Methode ist nicht auf das vorliegende Problem übertragbar, da keine unterschiedlichen Teilziele angestrebt werden. Der maximale Komfort soll immer angestrebt werden.
- (3) Bei den **Paarweisen Vergleichen** werden alle Ziele jeweils miteinander verglichen und das jeweils wichtigere Ziel bekommt einen Punkt. Da auch hier keine weitere Untergliederung Ziele gesetzt werden außer der maximale Anwenderkomfort, ist auch diese Methode nicht auf das vorliegende Problem anwendbar.
- (4) Eine weitere Methode ist die **Mittelwert-Kettungs-Methode**, bei der der Wertebereich von 0 bis 1 immer wieder halbiert wird, bis allen möglichen Entscheidungen ein Funktionswert zugeordnet ist. Diese Methode eignet sich ebenfalls nicht, um auf die Zufriedenheitsfunktion angewandt zu werden, da sie zum einen statisch arbeitet (und damit der Komplexität der vorliegenden Entscheidungen nicht gewachsen ist) und zum anderen sehr anfällig gegenüber sich ändernden Umgebungsbedingungen ist.

Von den genannten Methoden wird die Halbierungsmethode mit einer Rekursion gewählt, da damit alle Parameter bereits analytisch bestimmt werden können.

$$a(c) = -c \tag{57}$$

Zur Ermittlung dieser analytischen Lösung in Abhängigkeit von allen Randbedingungen wurde die Funktionsgleichung aus (56) nach den Variablen a und b , jeweils in Abhängigkeit von c umgestellt, womit sich (57) und (58) ergeben.

$$b(c) = \ln\left(1 - \frac{500}{c}\right) \quad (58)$$

Jetzt kann man die konkave Form über den Parameter c steuern. Dieser kann im vorliegenden Fall den Definitionsbereich $c \in [-\infty; 0)$ liegen, wobei sich ein $c \rightarrow 0$ in einer sehr großen Krümmung (eine Abweichung im nahezu perfekten Komfortbereich hat sehr viel mehr Einfluss als im schlechten Komfortbereich) und $c \ll 0$ eine fast lineare Abbildung von objektivem Wert und Nutzerzufriedenheit zu Folge hat. In Abbildung 30 ist diese Abhängigkeit für verschiedene Werte von c dargestellt.

Um $s_{\text{kor}}(t)$ für die halbe Nutzerzufriedenheit $f_{\text{comfort}}(s_{\text{kor}}, 0,5) = 250$ zu ermitteln, wurden Befragungen im persönlichen Umfeld durchgeführt sowie Artikel über Smart Home in diversen Fachzeitschriften herangezogen sowie das eigene Empfinden interpretiert, da in der Literatur keine repräsentativen Daten dazu gefunden wurden.

Die halbe Zufriedenheit entspricht danach einem Wert von 250. Das bedeutet, wenn der Anwender bei 70 % korrekt angesteuerten Geräten bereits halb zufrieden ist, muss $c = -100$ gewählt werden, weil diese Kurve bei 0,7 (70 %) den Funktionswert von 250 erreicht. Da dem Anwender bei mindestens 83 % korrekt geschalteter Geräte (bei längerer Korrekturzeit mehr als 83 %) eine halbe Zufriedenheit unterstellt wird, wird $c = -10$ gewählt.

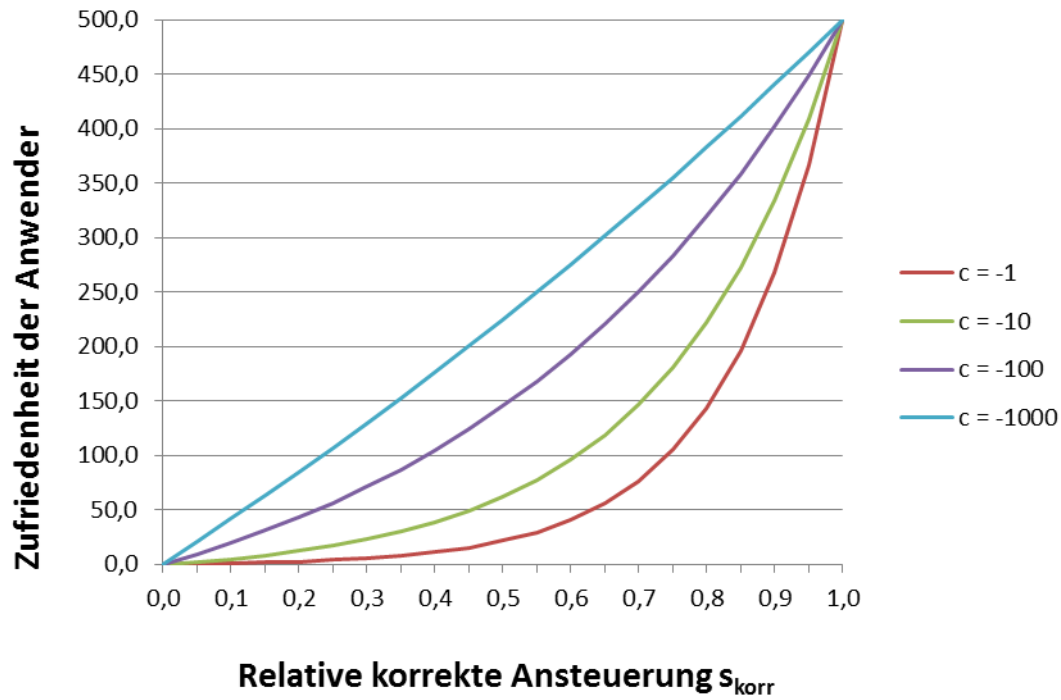


Abbildung 30. Abhängigkeit der Zufriedenheit vom Parameter c.

Damit ergibt sich die subjektivierte Zufriedenheitsfunktion zu (59).

$$f_{\text{comfort}}(t) = k_{\text{Norm}} (10 \bullet e^{3.9318s_{\text{korr}}} - 10) \quad (59)$$

Um als letzten Schritt zur Bildung der Teilzielfunktion Komfortmaximierung den Normierungsfaktor k_{Norm} zu berechnen, ist die Anzahl der simulierten Halbstunden wichtig. Damit ergibt sich (mit jeweils 48 Halbstunden h') für 12 Monate im Jahr k_{Norm} , wie in (60) gezeigt ist.

$$k_{\text{Norm}} = \frac{1}{12 \frac{\text{Monat}}{a} \bullet 48 \frac{h'}{\text{Monat}}} = \frac{1}{576} \frac{a}{h'} \quad (60)$$

Die Visualisierung der Teilzielfunktion $f_{\text{comfort}}(s_{\text{korr}})$ ist in Abbildung 31 gezeigt. Der Funktionswert ist zunächst einheitslos, da es keine Einheit zur

Beschreibung der Zufriedenheit gibt. Der Funktionsbereich erstreckt sich von $s_{\text{kor}}(t) = 0$ bis $s_{\text{kor}}(t) = 1$, womit sich der Wertebereich in Tabelle 6 ergibt (der Maximalwert für $f_{\text{comfort}}(s_{\text{kor}})$ ergibt sich aus dem Wertebereich von $f_{\text{comfort}}(s_{\text{kor}}) = [0 \dots 500]$ und insgesamt 576 Halbstundenintervalle).

$s_{\text{kor}}(t)$	$f_{\text{comfort}}(s_{\text{kor}})$	Interpretation
0	0,000	Kein Gerät wird korrekt angesteuert
1	0,868	Alle Geräte werden korrekt angesteuert

Tabelle 6. Wertebereich der Funktion $f_{\text{comfort}}(s_{\text{kor}})$.

Die grafische Darstellung in Abbildung 31 zeigt die Abhängigkeit der Zufriedenheit von der korrekten Ansteuerung der Geräte durch ENKOS im möglichen Funktionsbereich.

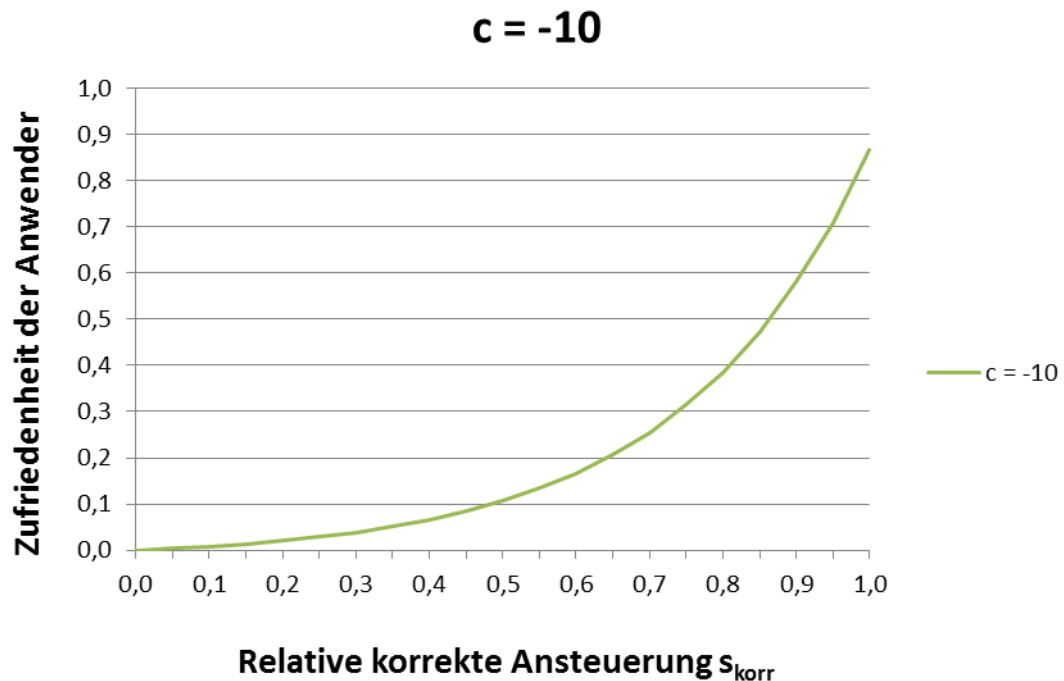


Abbildung 31. Abhängigkeit der Zufriedenheit vom Parameter c (Halbstundenintervall).

4.3.1 Methoden für Teilziel Komfortmaximierung

Das Teilsystem für die Komfortsteigerung, ausgehend von der Theorie der Entscheidungsfindung, ist eine Kernkomponente von ENKOS, die den Anwenderwunsch bestmöglich adaptieren soll, da die Nutzerzufriedenheit sich direkt in den Nutzerkomfort überführen lässt. Dafür wird als gedankliche Grundlage die menschliche Entscheidungsfindung gewählt, die die Generierung des Sollzustands s'_{tl} (vgl. Abbildung 47) durchführt, indem er die aktuelle Situation empfängt und als Folge seiner Entscheidungen die Verbraucher im Haushalt entsprechend steuert. Deshalb soll im ENKOS im Kasten „Komfortmaximierung“ eine Methode des Verhaltenserwerbs implementiert werden, da diese den menschlichen Entscheidungsprozess abbildet.

Deshalb wird zunächst festgelegt, dass eine Entscheidung genau einmal im Zyklus (Zeitschritt) getroffen wird. Zudem wird das System für individuelle Entscheidungen entwickelt, da für den Nachweis der Tauglichkeit das Verhalten eines Individuums nachgebildet werden soll [6].

Außerdem wird eine Entscheidung *logisch* eingegrenzt auf die Steuerung genau eines elektrischen Geräts, sodass ein Schaltvorgang von n_{Ger} Geräten auch n_{Ger} Entscheidungen benötigt.

Dabei sind die Instrumente der Generalisierung und Abstraktion wichtig, die aus beobachteten Beispielen (induktiv) auf generelle Regeln der individuellen Entscheidungsfindung schließen, um diese zur Anwendung zu bringen. Das Beherrschen dieser Regeln ist die Referenz für die Komfortmaximierung. Da erfahrungsgemäß auch die Umweltfaktoren für menschliche Entscheidungen eine Rolle spielen, sind diese ebenfalls durch ENKOS zu detektieren. Die angestrebte Methode des Verhaltenserwerbs führt automatisch eine Parameteridentifikation durch.

Als Sonderfall zur rein psychologischen Entscheidungssituation werden alle gefällten Entscheidungen vollständig umgesetzt, womit der Folgezustand s_{t1} im Sinne der Entscheidungstheorie beherrschbar ist. Dazu werden zunächst die Typen von Entscheidungstheorien auf deren Integrierbarkeit in dem geplanten System laut Abbildung 47 hin untersucht.

4.3.2 Das Modell eines Entscheidungsprozesses

In Abschnitt 6.3 wird der Entscheidungsprozess in ENKOS integriert und folgendermaßen beschrieben:

- (1) Im Rahmen der Identifikation wird der Bedarf einer Entscheidung erkannt, wenn sich der Umweltzustand (signifikant) s_{t0} ändert. Eine Änderung ist beispielsweise gegeben, wenn eine Person einen Raum betritt oder sich die Helligkeit über einen Diskretisierungsbereich hinaus ändert.
- (2) Alle möglichen Entscheidungsoptionen und deren zu erwartender Nutzen sind in der Wissensbasis von ENKOS vorhanden.
- (3) Die Entscheidung, die den größten Nutzen im Sinne eines längerfristigen Ziels erwarten lässt, wird getroffen nach dem Maximierungsprinzip.
- (4) Im Rahmen der Implementierung wird die Entscheidung (Intention) an die Akteure vermittelt, damit sich der gewünschte Zustand wirklich herstellt und durch den Anwender bewertet werden kann.
- (5) Zum Schluss erfolgt das Feedback, wobei die realisierte Aktion a_{t1} entweder durch den Anwender belohnt oder bestraft wird. Vorausgesetzt wird hierbei die Berechnung der Zielfunktion, womit das Feed-

back die entsprechende Gewichtung erfährt. Außerdem wird sie damit steuerbar, je nach den Bedürfnissen der Anwender.

Präselektionale Phase

Die ersten drei Phasen werden im Entscheidungsprozess als Präselektionale Phase definiert, da sie die Entscheidung an sich durch die Sammlung und Validierung von Informationen vorbereiten. Nach der Nutzentheorie [61] steht die Suche nach der dominanten Entscheidungsoption mit dem größten subjektiv zu erwartenden Nutzen im Vordergrund, die wiederum zwei Voraussetzungen von ENKOS fordert, um eine vollständige Analyse durchzuführen:

- (1) Alle möglichen Kombinationen von s_{it} , und deren Konsequenzen für den Anwender, also die Belohnung bzw. Bestrafung in Form von r_{it} , sollten bekannt sein. Bei der Anzahl von n_{Ger} Geräten handelt es sich um $2^{n_{Ger}}$ Möglichkeiten (bei zwei möglichen Schaltzuständen).
- (2) Eine für alle Entscheidungen anwendbare Nutzenfunktion muss existieren, die den objektiven Wert einer Entscheidung in den subjektiven Wert¹⁹ einer Konsequenz abbildet. In (61) wird diese Umrechnung gezeigt, wobei die Eintrittswahrscheinlichkeiten p_i als 1 gesetzt werden, da jede getroffene Entscheidung im vorliegenden Smart Home-Ansatz auch realisiert wird (s. Schritt 4 in Abbildung 17).

¹⁹ Also den Wert bezogen auf die Person (den Nutzer), womit aus dem gesamten Nutzen die Nutzersicht interpretiert und bewertet wird.

$$EU = \sum u_i \cdot p_i \stackrel{p_i=1}{=} \sum u_i \quad (61)$$

Die Nutzenfunktion $EU(u,p)$ kann, je nach Entscheidungsproblem, einen linearen, einen konvexen oder einen konkaven Verlauf haben. In der vorliegenden Aufgabenstellung muss der empfundene Nutzen eines Smart Home-Anwenders damit beschrieben werden. Bei risikoaversen Entscheidungssituationen wird der konkave, bei risikofreudigen Entscheidungen der konvexe und sonst der lineare Kurvenverlauf benutzt.

Selektionale Phase

Die vierte Phase ist die Selektionale Phase, in der die Entscheidung selbst getroffen und damit die Intention zur Veränderung der Umwelt im Sinne des Entscheiders geäußert wird. Um zu erklären, wie genau eine Entscheidung aufgrund der gesammelten Informationen gefällt wird, gibt es verschiedene Ansätze, unter denen sich die Prospect Theory weitgehend durchgesetzt hat. Der erwartete objektive Wert EU wird anschließend durch die Wertfunktion (hervorgehend aus der Nutzenfunktion) zu SEU subjektiviert, bevor dieser wiederum mit allen Alternativen verglichen wird. Aus der Erfahrung heraus ist die Umrechnung eine logarithmische Funktion.

Postselektionale Phase

Die Phasen fünf und sechs fassen die Postselektionale Phase zusammen, wo die Entscheidung umgesetzt und das Feedback auf diese Entscheidung eingeholt wird. Das Gedächtnis soll in ENKOS selbst abgebildet werden.

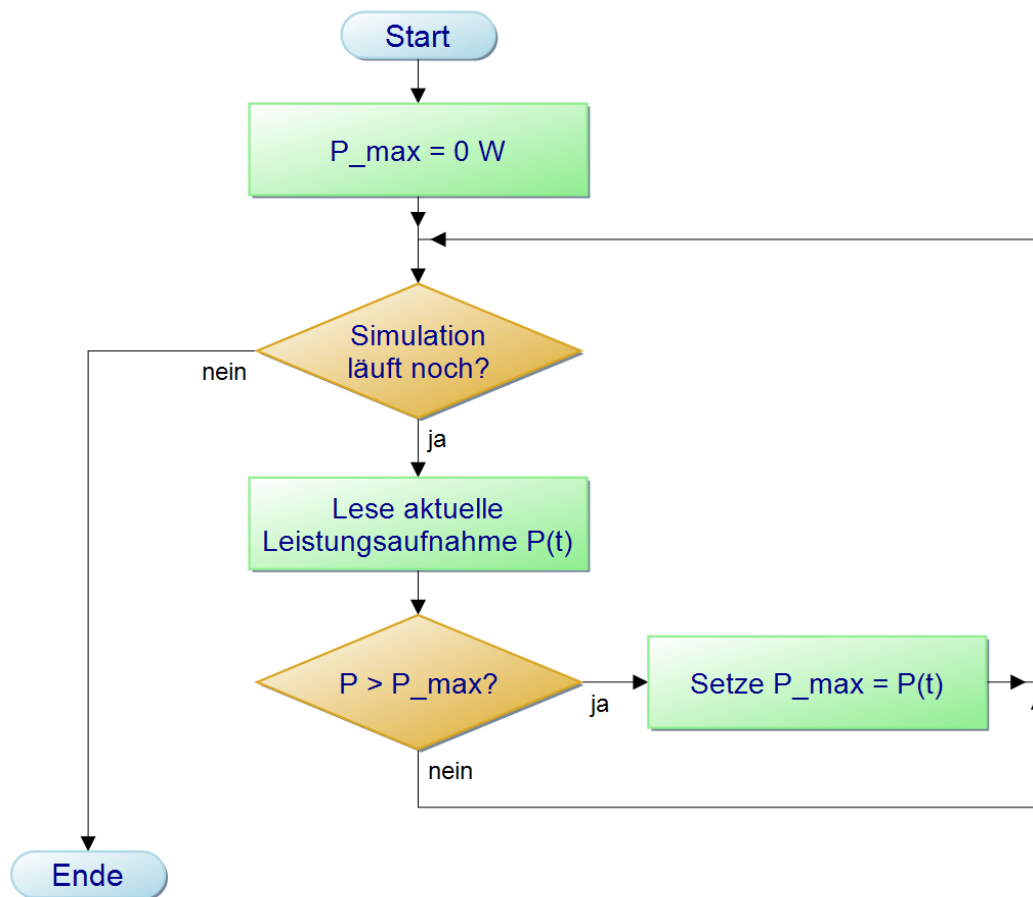
4.4 Definition der Teilzielfunktion $f_{\text{energy}}(t)$ für die Energieminimierung

Im nächsten Schritt wird die Funktion f_{energy} in Abhängigkeit der eingelesenen Sensorfunktion, mit deren Hilfe die Leistungsaufnahme erfasst wird, definiert. Bei diesem Teilziel soll der Energieverbrauch in einem Haushalt durch die Anwendung geeigneter Methoden, die später untersucht werden, im Rahmen einer zentralen Steuerung aller Geräte durch ENKOS verringert werden. Als erstes werden wiederum die Kurvenform und danach die genauen Parameter für die Funktion festgelegt.

Da die Belohnung direkt proportional zum elektrischen Energieverbrauch sein soll und eine Subjektivierung unnötig ist, wird als prinzipielle Kurvenform eine Gerade erster Ordnung festgelegt, für deren Parameterbestimmung zwei Arbeitspunkte definiert werden müssen. Ein Arbeitspunkt der Abbildung des Funktionswertes in Abhängigkeit vom tatsächlichen Energieverbrauch dient dabei ein Verbrauch von $E_{\min} = 0$ kWh.

Dieser Verbrauch soll maximal belohnt werden, womit sich dieser Arbeitspunkt für ein ganzes Jahr bei $f_{\text{energy}}(E = 0 \text{ kWh}) = 500$ ergibt. Der zweite Arbeitspunkt leitet sich aus dem maximal möglichen Energieverbrauch über das Normaljahr hinweg ab, der die Belohnung 0 erhalten soll. Dabei muss die maximale Leistungsaufnahme P_{\max} aller elektrischen Geräte ermittelt

werden, indem dieser über einen bestimmten Zeitraum ermittelt wird. Das ist ein rekursiver Prozess, wie in Prozess 2 dargestellt.



Prozess 2. Rekursive Ermittlung des zweiten Punktes P_{\max} für $f_{\text{energy}}(P(t))$.

In der simulierten Wohnung ergab sich nach Ausführung dieses Prozesses eine maximale elektrische Leistungsaufnahme aller Geräte von $P_{\max} = 1510 \text{ W}$, womit sich $E_{\max} = 755 \text{ Wh}$ für ein Halbstundenintervall h' berechnet. Dieser Wert kann einfach in den maximal mögliche Energieverbrauch umgerechnet werden, da in der Simulation jede halbe Stunde ($\Delta t = 1 \text{ h}' = \frac{1}{2} \text{ h}$) eine Abtastung erfolgt und die zwischen zwei Abtastungen liegende Leistungsaufnahme linear interpoliert wird.

Deshalb wird dieser Wert E_{\max} als fester Wert für den maximalen Misserfolg wenn also permanent die maximale elektrische Leistung angenommen, der mit $f_{\text{energy}}(E_{\max}) = 0$ bestraft wird. Als zweite Randbedingung ergibt sich die maximale Belohnung bei einem Energieverbrauch von $E(t) = 0$ Wh. Aufgrund der Skalierung der Teilfunktion auf einen Wertebereich von $f_{\text{energy}}(E(t)) = [0 \dots 500]$ (vgl. Abschnitt 4.2 und (47)) und der Einteilung des Jahres in 576 h^e ergibt sich eine maximal mögliche Belohnung von 0,868 für jedes Halbstundenintervall.

E(t) in [Wh]	$f_{\text{energy}}(E(t))$	Interpretation
0	0,868	Minimale elektrische Energieaufnahme
755	0,000	Maximale elektrische Energieaufnahme

Tabelle 7. Wertebereich der Funktion $f_{\text{energy}}(E(t))$ im Simulationsbeispiel.

Die Form der Funktion wird als linear angenommen, da sich die Zufriedenheit genau umgekehrt zum auszugebenden Geld und damit zum Energieverbrauch verhält. Also ergibt sich diese Teilfunktion im Allgemeinen zu (62).

$$f_{\text{energy}}(E(t)) = k_{\text{Norm}} \left(500 - 500 \frac{E(t)}{E_{\max}} \right) = 500 k_{\text{Norm}} \left(1 - \frac{E(t)}{E_{\max}} \right) \quad (62)$$

Als letzter Schritt muss auch hier der aktuelle Wert zu jedem halbstündlichen Zeitschritt normiert werden, womit k_{Norm} auch in dieser Teilfunktion hinzumultipliziert wird. Die graphische Darstellung ist in Abbildung 32 gezeigt.

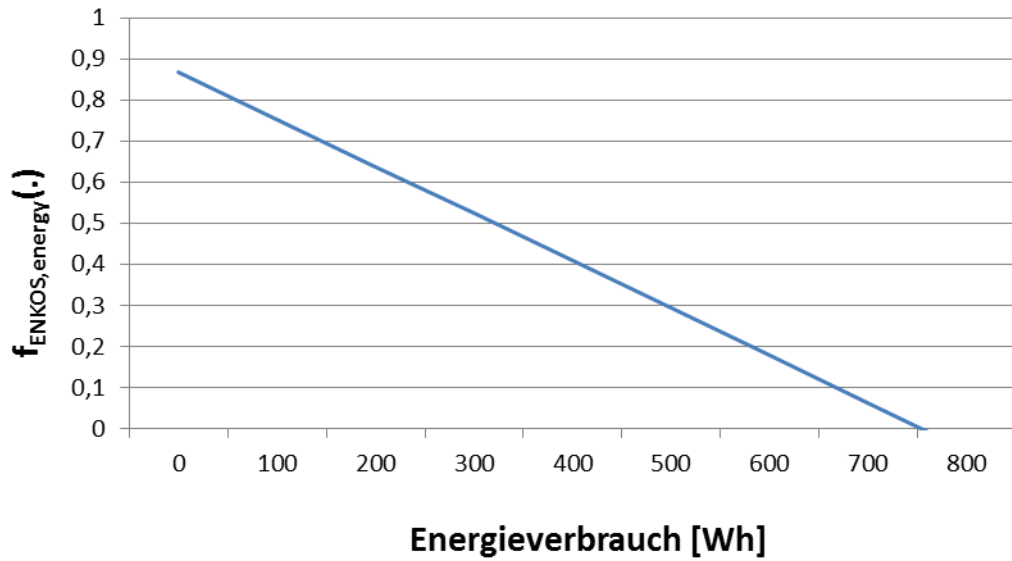


Abbildung 32. Abhängigkeit der Zufriedenheit vom Energieverbrauch.

Die Entscheidung des Systems, wann ein Gerät tatsächlich eingeschaltet werden sollte, ergibt sich in Abschnitt 6.4, da ein Verbrauch von 0 Wh in vielen Anwendungsfällen zwar die in diesem Abschnitt behandelte Teilfunktion $f_{\text{energy}}(t)$ maximiert, für die andere Teilfunktion $f_{\text{comfort}}(t)$ jedoch sehr niedrige Werte erzeugt (beispielsweise wenn Personen im Raum sind, die das Licht aufgrund der Dunkelheit eingeschaltet haben möchten).

4.5 Aufstellen der Zielfunktion f_{ENKOS} aus den Teilzielen

Mit den beiden Teilzielfunktionen ergibt sich die Zielfunktion des Systems beim Zeitpunkt $t = t_0$ (vgl. (63)). In (60) ist k_{Norm} bereits definiert worden.

$$f_{\text{ENKOS}}(t) = k_{\text{Norm}} \left(10 \bullet e^{3,9318 s_{\text{kor}} - 10 + 500 \left(1 - \frac{E(t)}{E_{\text{max}}} \right) \right) \quad (63)$$

Die Funktion f_{ENKOS} hat damit einen Wertebereich von $[0, 1,736]$, wobei $f_{\text{ENKOS}} = 0$ völlige Unzufriedenheit des Anwenders und $f_{\text{ENKOS}} = 1,736$ völlige Zufriedenheit (minimaler Energieverbrauch) widerspiegelt. Die Darstellung des Funktionswertes abhängig von beiden Eingangsparametern $s_{\text{korrt}}(t)$ und $E(t)$ ist in Abbildung 33 gezeigt.

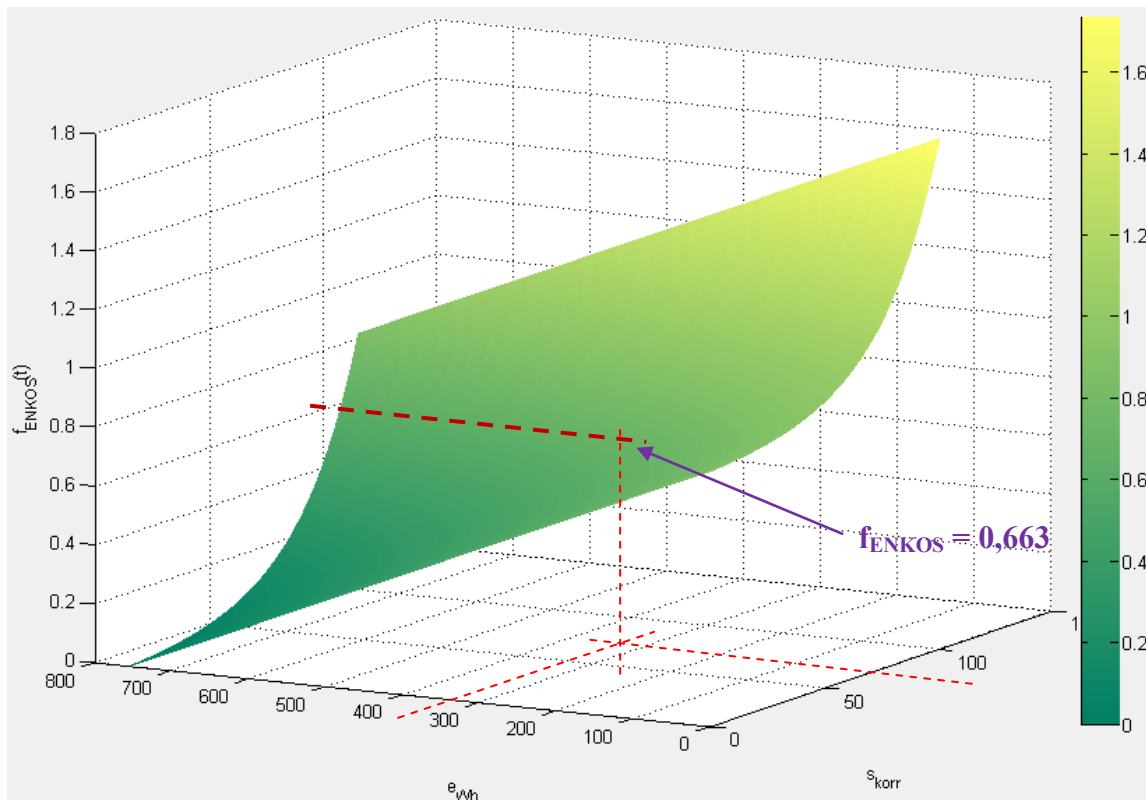


Abbildung 33. Abhängigkeit der Zufriedenheit vom Energieverbrauch (Zielfunktion).

Diese Zufriedenheit für einen gegebenen Zustand, bei dem 70 % aller Geräte korrekt (im Sinne des Nutzers) geschaltet sind ($s_{\text{korr}}(t) = 0,7$) und der Energieverbrauch $E(t) = 400$ Wh beträgt, ergibt sich $f_{\text{ENKOS}} = 0,663$. Dieser Punkt ist in Abbildung 33 beispielhaft visualisiert. zu jedem Zeitpunkt wird zur Bewertung der aktuell getroffenen Entscheidung herangezogen, wenngleich eine Bewertung des gesamten Normaljahres den Algorithmus von ENKOS mit den gewählten Parametern an sich bewerten soll. Damit ist eine Auf-

summierung von allen 576 simulierten Zeitpunkten nötig, was in (64) gezeigt ist. Der Wertebereich reicht hier von 0 (völlige Unzufriedenheit über das gesamte Jahr) bis 1000 (völlige Zufriedenheit).

$$f_{\text{ENKOS}} = \sum_{t=0}^{576 \text{ h}'} f_{\text{ENKOS}}(t) \quad (64)$$

Mit der Definition dieser Zielfunktion sind nun eine objektive Vergleichbarkeit der strukturellen Eigenschaften des Modellansatzes sowie verschiedener Modellparameter gewährleistet.

5 Die Simulation des Standardhaushalts und die Entwicklung von ENKOS

Der Standardhaushalt für die Simulation umfasst sämtliche Geräte in zwei Zimmern im eigenen Wohngebäude, und zwar des Arbeitszimmers und des Bads, sowie einige Geräte außerhalb dieser Zimmer. Eine Übersicht findet sich später in Tabelle 9. In den Zimmern ist die Anwesenheit von Personen ermittelt worden, womit eine Zuordnung „Anwesenheit – Gerät einschalten“ für die Geräte im Arbeitszimmer und im Bad möglich ist.

5.1 Grundsätzliche Versuche im eigenen Haus

Im Rahmen von grundsätzlichen Untersuchungen soll die Machbarkeit einer einfachen Steuerung im Sinne eines Smart Home-Ansatzes einer selbst entwickelten Hardware ermittelt werden. Das wurde im eigenen Haus²⁰ durchgeführt. Dabei wurde im ersten Schritt eine einfache Topologie aus einem Sensor zur Anwesenheitserkennung in einem Raum sowie der Steuerung eines Verbrauchers aufgebaut.

Die Detektion der Anwesenheit wurde durch ein Lichtschrankenpaar realisiert, dessen Integration in die Eingangstür in Abbildung 34 als Skizze und in Abbildung 35 als räumlicher Aufbau gezeigt ist. Dabei wird ein Infrarotlichtstrahl ausgesandt und auf der gegenüberliegenden Seite durch zwei Sensoren detektiert. Da sich damit mit einem Sender gleich zwei Lichtschranken realisieren lassen, wird nachfolgend vom Lichtschrankenpaar gesprochen.

²⁰ Adresse: Ausfeldstr. 21, 99880 Waltershausen

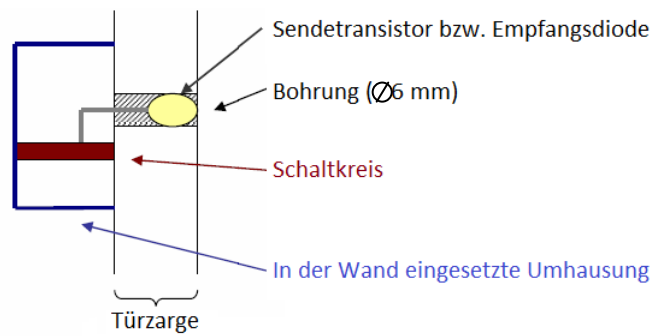


Abbildung 34. Einbau der Lichtschranke (Sender) in eine Türzarge. Eine Sendediode versorgt zwei Empfängertransistoren (zur Richtungserkennung) mit IR-Licht.

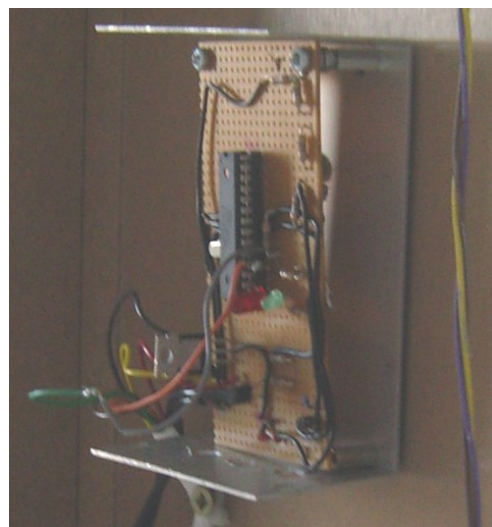
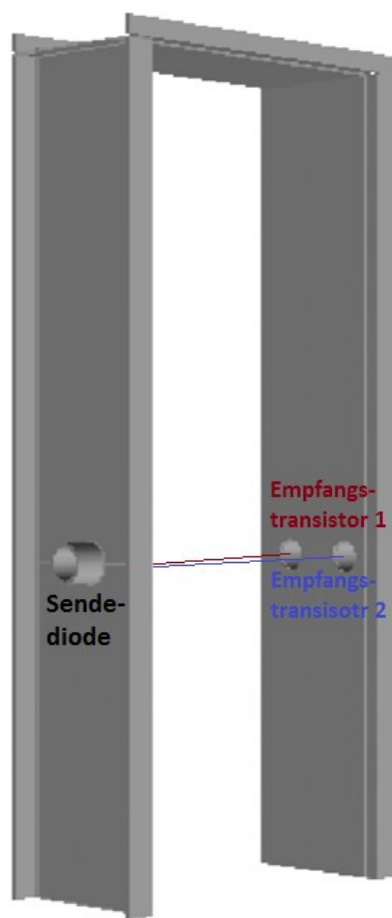


Abbildung 35. Einbau der Lichtschranken als Skizze (links) und praktische Ausführung am Beispiel der Empfangsdiode (rechts).

Dieses Lichtschrankschrankenpaar ist in Abbildung 35 gut zu sehen, in dem die rot-braune Linie den Weg des einen Lichtstrahls und die blaue den Weg des anderen darstellt. Beide werden vom selben Sender versandt.

Nach ersten Experimenten mit verschiedenen Personen (Kinder und Erwachsene) wurde bei einer Standardtür die Höhe des Lichtschrankschrankenpaars auf $h_{LS} = (1 \pm 0,05)$ m festgelegt, um ein zuverlässiges Schalten zu ermöglichen. Desweiteren wurden verschiedene Dioden sowie verschiedene Phototransistoren als Empfänger ausprobiert, bis die Wahl auf die LD 274 GaAs-IR-Lumineszenzdiode (Fa. OSRAM) auf der Empfänger- und den IC U2531B (Fa. Telefunken electronic) auf der Sendeseite fiel. Diese Kombination erwies sich als die robusteste.

Als nächstes musste durch die Einstellung eines Potentiometers eine Trägerfrequenz gewählt werden, damit die Einflüsse von anderen Lichtquellen, also des Sonnenlichts und der Hausbeleuchtung, keinen Einfluss auf das Signal haben. In einer Versuchsreihe mit den Trägerfrequenzen von 9,62 bis 18,91 kHz, die in Tabelle 8 dargestellt sind, wurde dabei die Versuchstür mehrfach in verschiedenen Geschwindigkeiten und von mehreren Personen zu unterschiedlichen Tageszeiten durchquert. Die größte Trägerfrequenz von 18,91 kHz erwies sich als diejenige mit der geringsten Störanfälligkeit.

Potentio- meter [k Ω]	0	1	2	3	4	5	6	7	8	9	10
Perioden- dauer [μ s]	52	57	63	68	73	78	83	89	94	99	104
Trägerfre- quenz [kHz]	19	17	16	15	14	13	12	11	10	10	9

Tabelle 8. Verschiedene Trägerfrequenzen wurden ausprobiert bezüglich Robustheit des Lichtschrankenspaars gegenüber Fremdlichteinflüssen.

Die Auswertung der Personenbewegung ins Zimmer hinein bzw. aus dem Zimmer heraus erfolgte durch einen Mikrocontroller Atmel Mega 8, der im C-Code programmiert wurde. Das Grundgerüst des Programms bildete eine Zustandsmaschine, die in eine Endlosschleife integriert wurde. Die interruptbasierte Lösung erwies sich als nicht stabil genug, da eine eindeutige Priorisierung der Interrupts hier nicht möglich war. In den Tests wurden durch das Prellen des Eingangssignals Interrupts immer wieder ausgelöst, wodurch die wesentlichen Signale teilweise nicht mehr erfasst wurden. Eine Entprellung war aufgrund der begrenzten Ressourcen nicht möglich, und ein größerer Mikrocontroller fiel für eine solch einfache Aufgabe des Zählens anwesender Personen nicht in Betracht.

Die ebenfalls selbst entwickelte Schaltung, die die Lichtschrankensignale erfasst, aufbereitet und dem Mikrocontroller zur Verfügung stellt, ist in Abbildung 36 gezeigt. Auf der linken Seite der Abbildung befinden sich die Spannungsregler für die 5 V-, 9 V- sowie die 12 V-Versorgung, während in der Mitte der Controller sowie dessen Treiberbeschaltung zu sehen sind. Dieser empfängt die Signale der Lichtschrankens auf den Kanälen INT0 und INT1 (siehe Markierung) und visualisiert per LED, wieviele Personen sich im Raum befinden. Die Anzahl der eingeschalteten LEDs entspricht die An-

zahl der Personen, wenn keine leuchtet ist auch keine Person anwesend. Desweiteren ist ein galvanisch entkoppelter Ausgang vorhanden, um die Schaltung mit anderen Schaltkreisen sowie einem PC per RS232-Schnittstelle zu verbinden. Damit wurde eine logische Master-Slave-Topologie realisiert, bei der alle Schaltungen als Slaves ihre aktuellen Anzahlen von Personen an eine Masterschaltung, und zwar der erste Schaltkreis, der sich im Netzwerk identifiziert, übermitteln und andererseits von dieser ihre Zeitstempel oder Resetbefehle empfangen. Damit ist eine Synchronisierung sichergestellt. Die Masterschaltung kommuniziert dabei mit einem PC, der diese wiederum steuert und ausliest.

Auf dem PC wurde dabei mit Hilfe der Programmiersprache Delphi wiederum eine Client- sowie eine Serveranwendung programmiert. Delphi eignet sich zum schnellen Aufbau von Software mit guter Funktionalität, da hier bereits viele grafische Bausteine vorhanden sind, die rasch mit Quellcode zur Funktionalisierung ausgefüllt werden können. Die Server bauen dabei eine Verbindung (via RS232- oder USB-Schnittstelle) zur Masterschaltung auf und kommunizieren regelmäßig. Wenn keine Befehle übertragen werden, wird ein „Alive-Signal“ mit dem aktuellen Zeitstempel gesendet, um der Schaltung einerseits die Anwesenheit eines PCs zu signalisieren und andererseits die Synchronisierung zu ermöglichen.

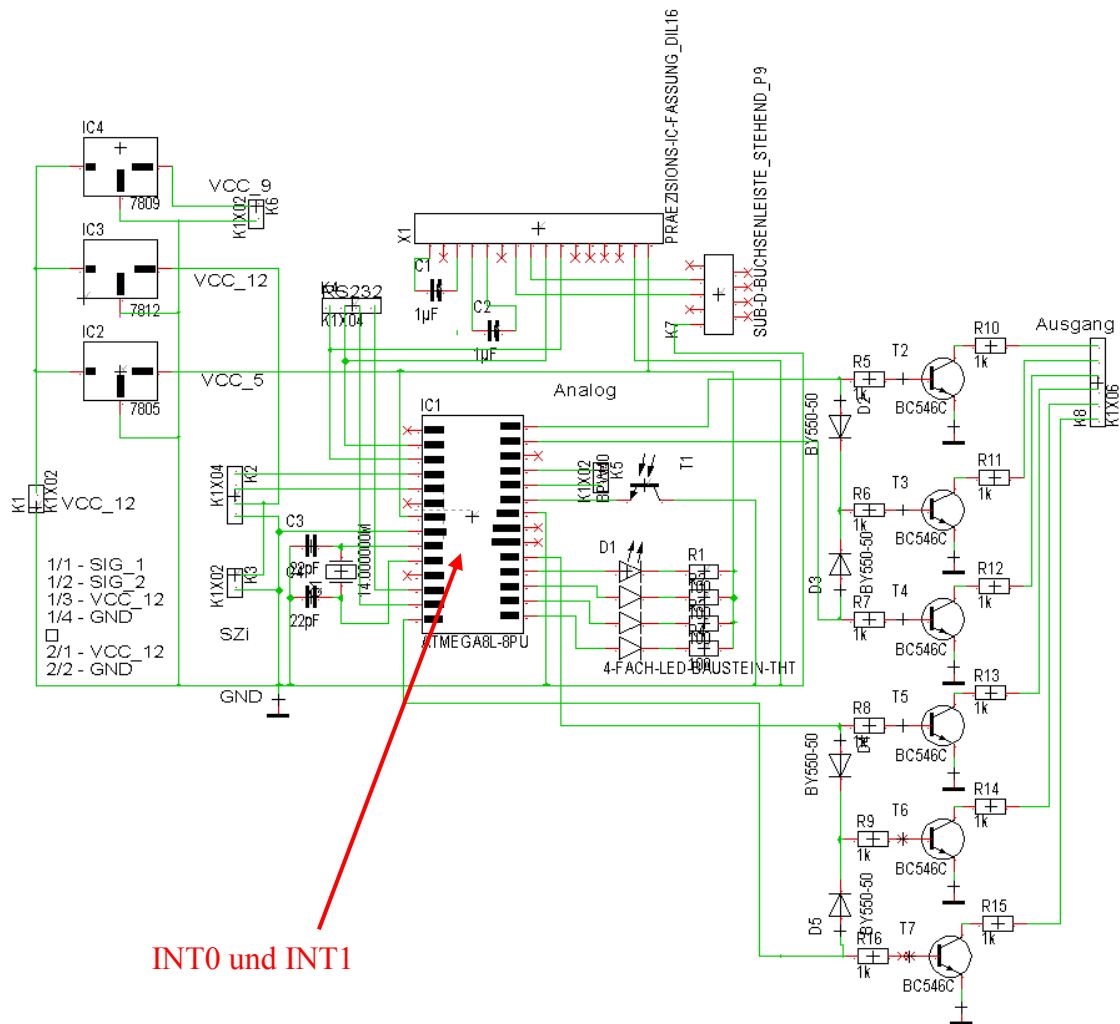


Abbildung 36. Schaltung zur Erfassung der Lichtschrankensignale.

Sobald der Client einen Befehl an die Serveranwendung sendet, wird sie dort übersetzt und an die Hardware weitergeleitet. Die Serveranwendung hat jederzeit den Überblick über alle Lichtschranken, da diese auf das Alive-Signal reagieren und sich zurückmelden. Damit ist sichergestellt, dass die Serveranwendung hinreichend viele Informationen über die Hardware hat, die sich dynamisch ändern kann.

Im Rahmen der Funktionspartitionierung wurden verschiedene Ansätze zur Speicherung der Informationen gegeneinander abgewogen. Die permanente, zyklische Aussendung eines Broadcast-Befehls durch einen PC, der wieder-

rum mit der Masterhardware verbunden ist, wurde verworfen. Denn hierbei muss permanent ein PC eingeschaltet bleiben, der den aktuellen Zustand anfordert, zurückliest und speichert.

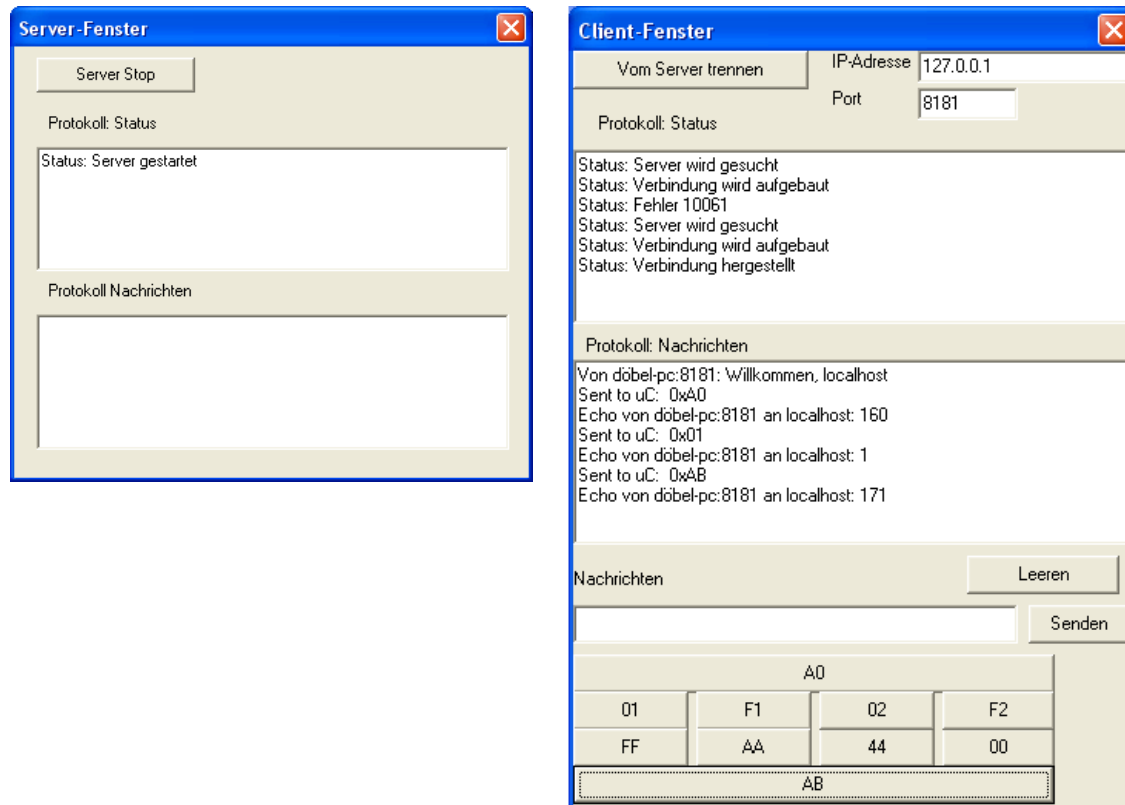


Abbildung 37. Client- und Serveranwendung auf dem PC als Delphi-Anwendungen.
Beide Anwendungen wurden selbst entwickelt und implementiert.

Deshalb ist als Alternative dazu eine dezentrale Speicherung jedes Ereignisses über mehrere Wochen hinweg (mit Systemzeitstempeln) in jeder einzelnen Schaltung (Slave) realisiert worden. Sobald die Serveranwendung auf dem PC gestartet wurde, meldet sich diese bei der Masterschaltung an und sendet einen Auslesebefehl. Damit ist die Historie bekannt. Das Kommunikationssequenzdiagramm ist im Anhang dargestellt, während sowohl die Client- als auch die Serveranwendung in Abbildung 37 gezeigt sind. Die Serveranwendung ist sehr ressourcenextensiv und wird während der gesam-

ten Laufzeit des PCs ausgeführt, der direkt über RS232 mit den Mikrocontrollern verbunden ist. Die eine oder mehrere Clientanwendungen hingegen können auf einem anderen PC installiert werden und verbinden sich auf einer logischen Ebene mit der Serveranwendung. Die Auswertung der permanent protokollierten Kommunikation wurde mit Excel automatisiert (s. Abbildung 38), während sich die selbst entwickelte Definition des Kommunikationsprotokolls im Anhang befindet.

<i>Physik. Datenübertragung</i>			<i>Empfangsbyte</i>	<i>Relativzeit</i>	<i>Relative Personenanzahl</i>	<i>Relative Helligkeit</i>
Sent	to uC:	0xA0				
22:08:01 - Antwort vom uC: 0xA0				20:29:12	33	
22:08:01 - Antwort vom uC: 0x1			1	00:00:00	0	0
22:08:01 - Antwort vom uC: 0x0			0			
22:08:01 - Antwort vom uC: 0x60			60			
22:08:01 - Antwort vom uC: 0x61			61			
22:08:01 - Antwort vom uC: 0x33			33			
22:08:01 - Antwort vom uC: 0x0			0			
22:08:01 - Antwort vom uC: 0x0			0			
22:08:01 - Antwort vom uC: 0x1			1	04:38:48	1	3
22:08:01 - Antwort vom uC: 0x0			0			
22:08:01 - Antwort vom uC: 0x63			63			
22:08:01 - Antwort vom uC: 0x1A			1A			
22:08:01 - Antwort vom uC: 0x34			34			
22:08:01 - Antwort vom uC: 0x0			0			
22:08:01 - Antwort vom uC: 0x3			3			
22:08:02 - Antwort vom uC: 0x1			1	05:50:24	2	3
22:08:02 - Antwort vom uC: 0x0			0			
22:08:02 - Antwort vom uC: 0x63			63			
22:08:02 - Antwort vom uC: 0xCD			CD			
22:08:02 - Antwort vom uC: 0x35			35			
22:08:02 - Antwort vom uC: 0x0			0			
22:08:02 - Antwort vom uC: 0x3			3			

Abbildung 38. Kommunikation mit automatisierter Auswertung. Im Beispiel ist auf der linken Seite die physikalische Übertragung gezeigt (7 Bytes).

Danach wurde in mehreren Versuchsreihen eine möglichst robuste Programmierung des Zustandsautomats auf den Mikrocontrollern gesucht. Auf-

grund der komplexen, mehrstufigen Toleranzkette der Eingangssignale, bestehend aus Umweltgegebenheiten (Lichteinstrahlung, Tageszeit), Hardware (Wahl der Trägerfrequenz, Fremdeinflüsse) sowie Software (äquidistante Abtastung) musste eine Plausibilisierung der eingelesenen Lichtschrankeninformationen durchgeführt werden, da die Summentoleranz zu ungültigen Zuständen führen konnte. So ist es beispielsweise nicht möglich, dass beide Lichtschranken LS_1 und LS_2 vom Zustand „0“ (kein Signal) gleichzeitig auf den Zustand „1“ (Signal erkannt) springen, sondern eine der beiden Lichtschranken muss zwingend zuerst schalten. In diesem Fall musste der nächste Zustand nach „ $LS_1=1$ & $LS_2=1$ “ abgewartet werden, um die Bewegungsrichtung der Person zu detektieren und damit den fehlenden Zustand zu ersetzen (vgl. Abbildung 39). Dazu wurden umfangreiche Testreihen mit einem PC aufgezeichnet, während einerseits Versuchspersonen (im eigenen Haushalt) durch die Tür liefen und andererseits Personen mit einem zweiten Mikrocontroller simuliert wurden.

Je nach eingefangenen Muster, also die Reihenfolge und Kombination, in der beide Lichtschranken unterbrochen werden, kann beurteilt werden, ob eine Person in den Raum hinein geht oder heraus kommt. Dabei sind alle gültigen Zustandsübergänge in Abbildung 39 gezeigt. Sowohl Start- als auch Endzustand sind jeweils Z_0 , während der Kreis entweder im Uhrzeigersinn oder entgegengesetzt durchlaufen wird. Je nach Anordnung der Lichtschranken bedeutet das jeweils einmal, dass eine Person das Zimmer betritt und das andere Mal das Heraustreten einer Person aus einem Zimmer.

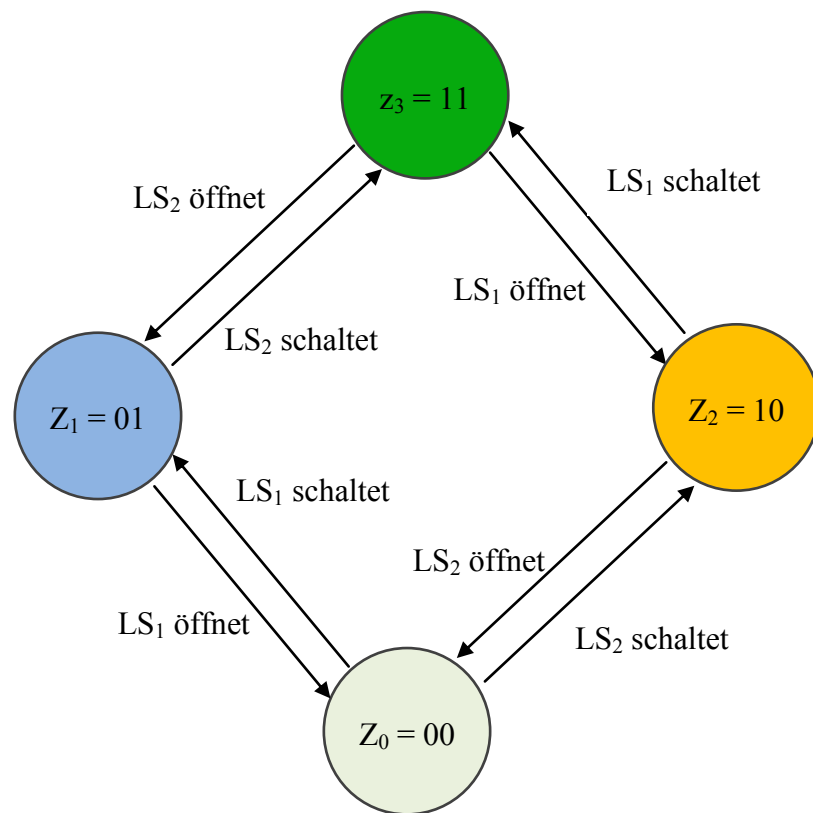


Abbildung 39. Zusammenfassung aller gültigen Zustandsübergänge.

Alle nicht gezeigten Zustandsübergänge sind theoretisch auch nicht möglich und wurden daher auf gültige Übergänge gemappt, sofern alle dafür nötigen Informationen zur Verfügung standen. Im Ergebnis konnte stabil erfasst und kommuniziert werden, wenn eine Person ein Zimmer betritt oder austritt. Die intern gemessenen Zustandsübergänge im Rahmen eines Versuchs sind in Abbildung 40 dargestellt. Der Kreis in Abbildung 39 wird im gezeigten Beispiel einmal gegen den Uhrzeigersinn durchlaufen.

16:18:40 - Antwort vom uC: 0xDD	0xDD		
16:18:40 - Antwort vom uC: 0xE2	0xE2	11100010	1 1 1 0 0 0 1 0 100010
16:18:40 - Antwort vom uC: 0xDD	0xDD		
16:18:40 - Antwort vom uC: 0x8B	0x8B	10001011	1 0 0 0 1 0 1 1 001011
16:18:40 - Antwort vom uC: 0xDD	0xDD		
16:18:40 - Antwort vom uC: 0x2D	0x2D	00101101	0 0 1 0 1 1 0 1 101101
16:18:41 - Antwort vom uC: 0xDD	0xDD		
16:18:41 - Antwort vom uC: 0xB4	0xB4	10110100	1 0 1 1 0 1 0 0 110100
16:18:41 - Antwort vom uC: 0x0	0x0		

Abbildung 40. Messung der internen Zustände des Mikrocontrollers der beiden Lichtschranken sowie das Mapping auf die farbige hinterlegten Zustände.

Nachdem sicher erkannt werden konnte, ob sich mindestens eine Person im Zimmer befindet, wurde eine Steuerung von Geräten in Abhängigkeit der Anwesenheit durchgeführt, indem das Mikrocontrollerprogramm entsprechend erweitert wurde. War plötzlich keine Person mehr anwesend, wurden die angeschlossenen Lampen durch eine handelsübliche Funksteckdose mit Fernbedienung der Firma Centor Warenhandels GmbH (Köln) abgeschaltet, indem die Funkfernbedienung durch den Mikrocontroller geschaltet wurde, wie Abbildung 41 und Abbildung 42 gezeigt ist.

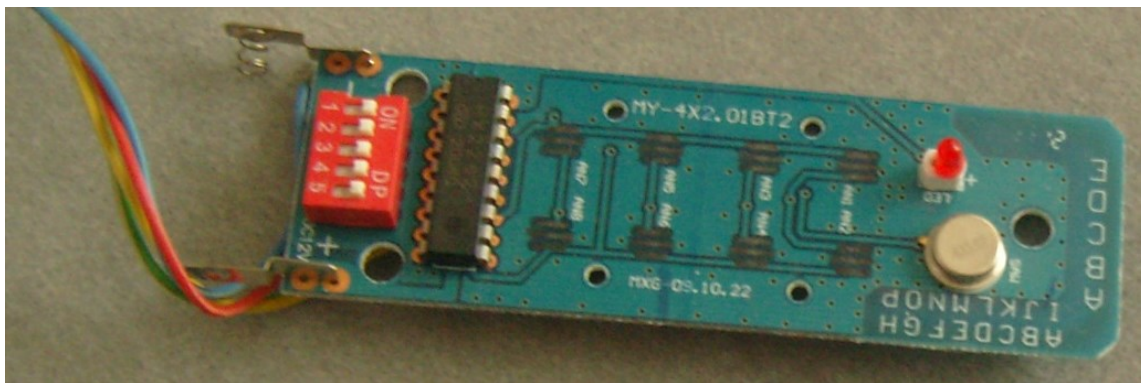


Abbildung 41. Verkabelung der Funkfernbedienung für die Steuerung der Geräte.

Dabei wurden die Knöpfe, die normalerweise manuell durch den Nutzer bedient werden, um Funksteckdosen ein- oder auszuschalten, durch den Controller „bedient“. Eine besondere Herausforderung stellte dabei die genaue

Ansteuerung dar. Es musste eine bestimmter Kurzschluss für eine bestimmte Zeitdauer im unteren Millisekundenbereich prellfrei hergestellt werden. Da die notwendige Zeitdauer jedoch von Versuch zu Versuch schwankte, wurden drei Impulse nacheinander mit den Zeitdauern von 5, 10 und 15 ms für einen einzigen Schaltvorgang gesendet. Damit konnte sicher gewährleistet werden, dass der Mikrocontroller die Steuerung übernimmt.

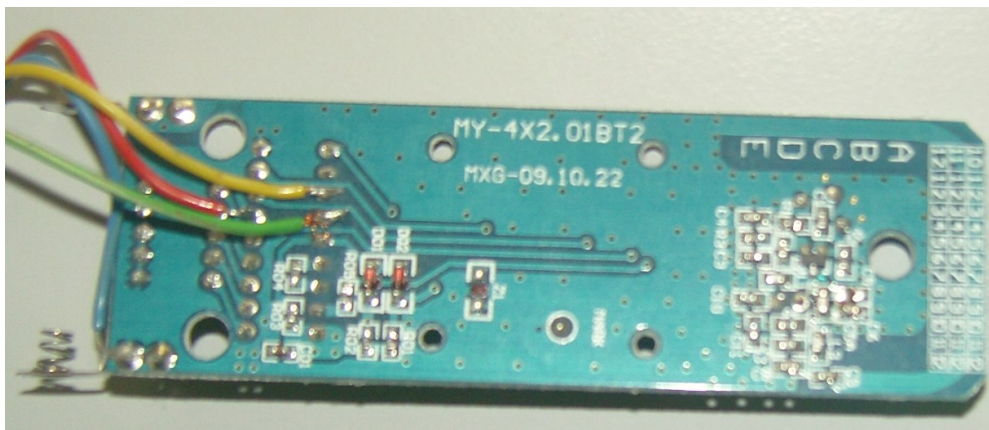


Abbildung 42. Verkabelung der Funkfernbedienung für die Steuerung der Geräte (Rückseite)

Sobald eine Anwesenheit detektiert wurde, stellte sich der ursprüngliche Einschaltwunsch der Geräte (Person war anwesend) wieder ein (s. Abbildung 43).

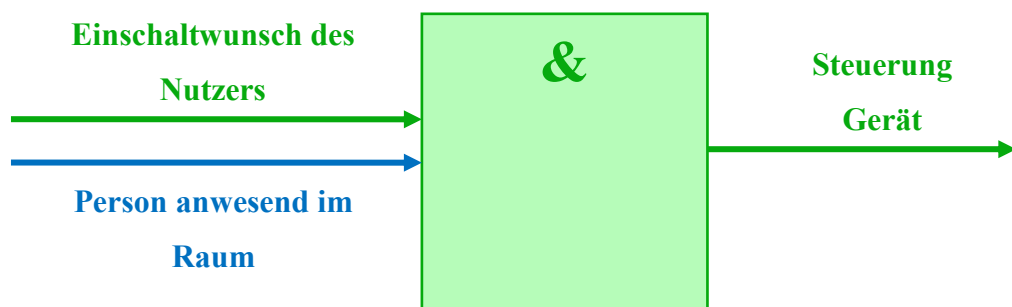


Abbildung 43. Einfache Steuerung in Abhängigkeit von der Anwesenheit einer Person.

Damit war die Einstellung des Zustands der elektrischen Geräte in Abhängigkeit von nicht steuerbaren Umweltbedingungen, und zwar der Anwesenheit, erfolgreich ausprobiert worden. Deshalb ist im Vorversuch ein möglicher Use Case der Anwesenheitserkennung hard- und softwaretechnisch umgesetzt worden.

5.2 Allgemeine Topologie der Simulationsumgebung

Als weitere Voraussetzung für die Entwicklung und Validierung eines Modellansatzes wurde neben dem Nachweis der Machbarkeit des Hardwareaufbaus eine Simulationsumgebung erstellt. Diese hatte zum einen die Aufgabe, einen deutschen Standardhaushalt abzubilden, also das Verbraucherverhalten in einer Wohnung, und zum anderen, die durch das entwickelte Modell getroffene Steuerentscheidung (stellvertretend für den Nutzer) quantitativ zu bewerten und damit sowohl $s_{\text{kor}}r$ als auch $E(t)$ für jeden Simulationsschritt. Der Standardhaushalt wurde in MS Excel umgesetzt, da hierbei eine tabellarische Eingabe des Haushalts leicht vorgenommen werden kann, während mittels Makros eine Kommunikation der Daten sowie der Steuerparameter für das Modell möglich sind.

In Abbildung 44 ist die generelle Topologie abgebildet. Die drei wesentlichen Bestandteile der Simulationsumgebung sind die 12 Normaltage, die jeweils das Verbraucherverhalten von einem Monat im Jahr repräsentieren, die Visualisierung aller Regeln inklusive deren Historie sowie das Konfigurationsmanagement.

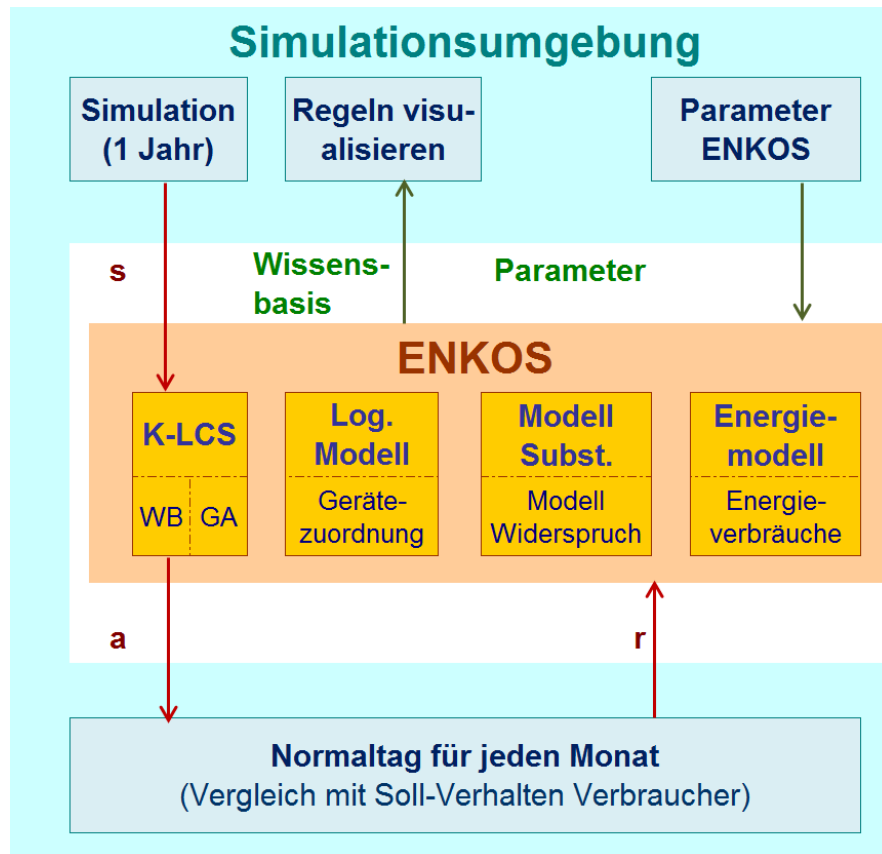


Abbildung 44. Die in Excel umgesetzte Simulationsumgebung. Diese besteht aus dem Treiber (oberer Teil der Simulationsumgebung) und dem „Stub“ (unterer Teil). Während der Treiber das Modell ENKOS als Dienst in Anspruch nimmt, stellt der Stub wiederum Dienste für ENKOS zur Verfügung. ENKOS selbst wird durch die Simulationsumgebung parametrisiert (WB = Wissensbasis, GA = Genetische Algorithmen).

Je nach Ausprägung von ENKOS soll eine Parametrisierung durch die Simulationsumgebung möglich sein. Exemplarisch ist in Abbildung 45 ein Ausschnitt des Normaltags für den Monat Januar dargestellt. In der linken Spalte ist die Kategorie der Verbraucher „Light“ ausgewählt, in der zweiten Spalte befinden sich alle Verbraucher dieser Kategorie. Anschließend sind die Leistungsaufnahmen in Spalte 3 gezeigt, die in der Simulation bereits vorgegeben sind. In der Praxis jedoch sind diese Werte gut messtechnisch zeitkontinuierlich zu ermitteln, beispielsweise durch das System von Digi-

talstrom, das mit einer Auflösung im Sub-Wattbereich sehr genau Leistungsaufnahmen aller Geräte ermitteln kann.

Anteil am Normmonat	Gerät	Uhrzeitintervall									
		19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	
		20:00	20:30	21:00	21:30	22:00	22:30	23:00	23:30	00:00	
		Lichtstärkeintervall [%]									
		0	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0	0
Person anwesend											
Arbeitszimmer	Lampe_1			x	x	x					
	Lampe_2					x	x				
	PC_1				x	x	x	x	x	x	
	PC_2 (mobil)				x	x	x				
	Tischlampe					x	x				
	Radio	x	x			x	x				
TV/ Audio	TV & Zubehör				x	x					
	Badradio					x	x				
	Radiowecker1		x	x							
	Radiowecker2						x	x			
	Audioanlage	x	x								
	Küchenradio					x	x				
Licht	Flur_1										
	Flur_2										
	Treppe	x									
	WoZi_1		x	x							
	WoZi_2										
	HWR		x								
	Außen										
	Küche	x									
	Bad										
	Spie_1		x	x							
	Spie_2		x								
	Dusche			x							
	Kind	x									
	Schlafzimmer				x						
	Arbeitszimmer				x	x					

Abbildung 45. Die Simulationsumgebung, die in Excel umgesetzt wurde.

Die nächsten beiden, grau hinterlegten Spalten sind Messwerte aus der Simulation, um einen rubrik- und zimmerweisen Abgleich mit den Daten des Bundesamts für Statistik [114] durchzuführen. Das bedeutet, dass die Messungen der Einschaltzeiten von Geräten im eigenen Heim hier hochskaliert wurden, bis sich der Energieverbrauch der Messungen mit denen des Bundesamts deckte, um den statistisch abgesicherten Daten das Muster des Selbstversuchs aufzuprägen.

Die hellen gelben Balken in Abbildung 45 stellen die statistisch angepassten Einschaltzeiten zum Zeitintervall, das in der oberen Zeile gezeigt ist, dar, was der Sollsituation s_{t_0} entspricht. Die dunklen gelben Balken sind die Antwort von ENKOS als Aktion a_{t_1} , die von der Simulationsumgebung gesendet und hier visualisiert werden. Damit ist ein direkter Soll-/ Ist-Vergleich bereits während der Simulation möglich.

5.3 Die Simulation eines Normjahres als Referenz

Ein Normaljahr dient als Referenzsimulation, an der ENKOS mit verschiedenen Parametern getestet wird. Damit wird das System ENKOS von Excel aus mit einem Zustandsvektor s versorgt und führt einen Berechnungszyklus durch. Sobald es eine Aktion vorgeschlagen hat, wird diese wiederum als Parametervektor a an Excel übertragen und vom virtuellen Nutzer bewertet. Nach der sich anschließenden Berechnung der Zielfunktion wird f_{ENKOS} als Reward an ENKOS geschickt, womit ENKOS sich selbst weiterentwickelt.

6 Modellbasierte Entwicklung von ENKOS

In den folgenden Abschnitten wird ein lernfähiger Gesamtalgorithmus entwickelt, um damit adaptiv den Zielfunktionswert f_{ENKOS} (vgl. (64)) zu maximieren. Dabei ist zunächst wichtig, die Topologie des Gesamtsystems und seiner internen und externen Schnittstellen aufzustellen, um danach geeignete Use Cases abzuleiten und geeignete Modellbildungsmethoden für die Komfortmaximierung wie auch für die Energieminimierung auszuwählen²¹. Als letzter Schritt sollen alle gefundenen Methoden in ein Gesamtkonzept integriert und dieses sowohl implementiert als auch verifiziert werden.

6.1 Allgemeine Topologie des Zielsystems und Einbettung in dessen Umwelt

Zunächst werden die Schnittstellen des Systems definiert. ENKOS soll als zentrale Steuerung für alle elektrischen Geräte im Haushalt arbeiten und Entscheidungen übernehmen, sobald ein bestimmter Modellreifegrad erreicht ist. Dieser ist technisch mit der Aussagesicherheit vergleichbar, also der mittleren Güte von vorhergesagten Entscheidungen eines Anwenders.

Desweiteren spielt die Generalisierungsfähigkeit des gesuchten Algorithmus‘ ebenfalls eine entscheidende Rolle zur Beschreibung des Reifegrads. Denn ein System mit einer großen Möglichkeit zur Ableitung von allgemeinen Gesetzmäßigkeiten aus konkreten Beobachtungen ist direkt proportional mit der Reife dieses.

²¹ Aufgrund des zu erwartenden unterschiedlichen Grundcharakters beider Ziele werden für beide jeweils eigene Ansätze gesucht, die nicht zwingend dem gleichen Typ entsprechen müssen.

Solange der Reifegrad für eigenständige Vorhersagen mit einer gewissen Treffsicherheit noch nicht erreicht wurde, befindet sich ENKOS in der *Lernphase* und die Wissensbasis wird aufgebaut (geführtes Lernen²²). Dabei wird der Zustand der elektrischen Geräte in der Wohnung durch den Anwender, analog einer Wohnung ohne intelligente Steuerung der Verbraucher, bestimmt. Diese Lernphase besteht aus den nachfolgend beschriebenen und in Abbildung 46 beschriebenen Schritten.

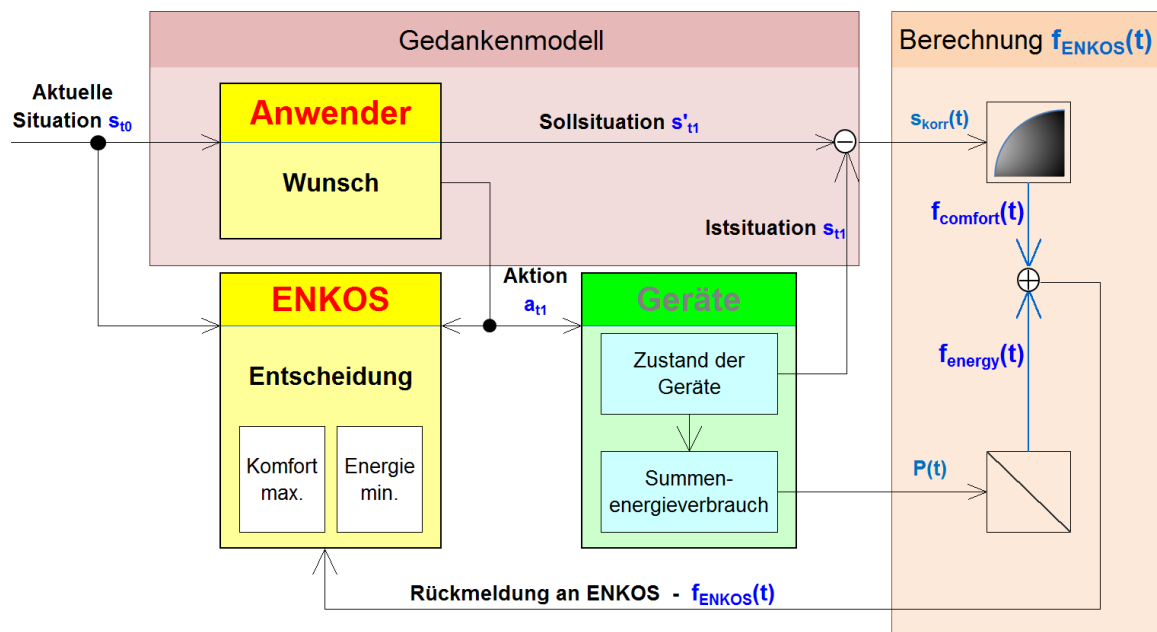


Abbildung 46. Systemtopologie während der Lernphase. Hier gibt der Anwender (Lehrer) die Entscheidung vor, mit der das Modell (ENKOS) in einer bestimmten Situation s_{t0} beaufschlagt wird.

²² In der Literatur wird in der Regel zwischen überwachten, nicht überwachten und bestärkenden Lernen unterschieden, wobei der Fokus dabei auf der Bildung des Lernmodells abhängig von einem vorhandenen Lehrer liegt. Von dieser Klassifizierung wird sich hier mit der Einführung des „geführten Lernens“ bewusst abgegrenzt, indem das Lernverfahren einerseits die Information über den Sollzustand (wie beim überwachten Lernen) in Ansätzen bekommt, andererseits die Konsequenz der Erzwingung des Sollzustands (als bestärkendes Lernen über die Zeitdauer bis zur Korrektur) als Maß für die Unzufriedenheit interpretiert wird.

Wie in Abbildung 46 zu erkennen ist, bildet ENKOS als Modell das Verhalten einer Person (als Wunsch a_{t1} , die Umwelt zu verändern, um s_{t1} einzustellen) in Abhängigkeit einer bestimmten Eingangssituation s_{t0} ab. Aufgrund der absehbaren Verschiedenheit der beiden Ansätze zur Umsetzung der Energieminimierung sowie der Komfortmaximierung wird bereits hier eine Partitionierung der Funktionalität in verschiedene Teilsysteme (Berechnung von $f_{\text{energy}}(t)$ sowie $f_{\text{comfort}}(t)$) vorgesehen.

Es ist jedoch nur ein Teil der Umwelt im Sinne des Verbrauchers schaltbar, und zwar die elektrischen Geräte und nicht etwa die Uhrzeit oder die Anzahl der Personen in einem Raum. Deshalb werden diese nachfolgend im Vektor der steuerbaren Umwelt δ_t zusammengefasst, während alle nichtsteuerbaren Elemente der Umwelt im Vektor ω_t enthalten sind. Beide zusammen bilden erst ein vollständiges Bild der Umgebung. Die Zuordnung der Geräte in steuerbare und nicht steuerbare ist in Tabelle 9 gezeigt. Der Einfachheit halber wird bei den Aktoren nur das (digitale) Ein- und Ausschalten betrachtet, während bei den Sensoren analoge Wertebereiche möglich sind.

Vektor	Elemente	Wertebereich
δ_t (Steuerbare Elemente)	Lampe 1 Arbeitszimmer	0: Gerät ist ausgeschaltet
	Lampe 2 Arbeitszimmer	
	PC 1 Arbeitszimmer	1: Gerät ist eingeschaltet
	PC 2 Arbeitszimmer	
	Tischlampe Arbeitszimmer	
	Radio Arbeitszimmer	
	TV	
	Badradio	
	Radiowecker 1	
	Radiowecker 2	
	Audioanlage	
	Küchenradio	
	Licht 1 Flur	
	Licht 2 Flur	
	Licht Treppenhaus	
	Licht 1 Wohnzimmer	
	Licht 2 Wohnzimmer	
	Licht HWR	
	Licht Außen	
	Licht Küche	
	Licht Bad	
	Licht Spiegelschrank 1	
	Licht Spiegelschrank 2	
	Licht Dusche	
	Licht Kinderzimmer 1	
	Licht Kinderzimmer 2	
	Licht Schlafzimmer	
	Licht Arbeitszimmer	
ω_t (nicht steuerbare Elemente)	Helligkeit	0 ... 100 %
	Uhrzeit (Halbstundenintervall h')	0:00 Uhr, 0:30 Uhr, ..., 23:30 Uhr
	Anzahl Personen im Bad	0 ... 5
	Anzahl Personen im Arbeitszimmer	(Absoluter Wert)

Tabelle 9. Auflistung aller Komponenten in der Wohnung, die in der Simulation aus Arbeitszimmer und Bad besteht, sowie deren Zuordnung in steuerbare und nichtsteuerbare.

Die Vektoren s_{t_0} , s'_{t_0} , s_{t_1} sowie a_{t_1} fassen die konkreten Messgrößen aus der Umwelt sowie die Steuergrößen an die Umgebung zusammen. Die genauen Zuordnungen sind in Tabelle 10 zusammengefasst. Während die blau ausgefüllten Felder bedeuten, dass δ_t bzw. ω_t ein Bestandteil des jeweiligen Vektors ist, zeigt das rot durchgekreuzte Feld an, dass die nicht steuerbaren Elemente der Umwelt auch kein Bestandteil des Aktionsvektors a_{t_1} sind. Eine mathematische Schreibweise, die aus dieser Tabelle resultiert, ist in (65) gegeben.

	s_{t_0} Sensierte IST- Situation zum Zeit- punkt $t = t_0$	s'_{t_0} Sensierte SOLL- Situation bei $t = t_0$	s_{t_1} Sensierte IST- Situation zum Zeit- punkt $t = t_1$	a_{t_1} Durchzufüh- rende Akti- on bei $t = t_1$
δ_t				
ω_t				

Tabelle 10. Komponenten der Vektoren s_{t_0} , s'_{t_0} , s_{t_1} und a_{t_1} . Der Zeitpunkt $t=t_0$ beschreibt den aktuellen Zeitschritt der Simulation, $t=t_1$ bedeutet bereits einen Schritt (also ein Halbstundenintervall) weiter. Die blau ausgefüllten Felder bedeuten die Zuordnung eines Vektors zu einem Vektor, das rote Kreuz ist keine Zuordnung.

Sobald eine neue Umweltsituation s_{t_0} , also eine Differenz zur aktuell sich im Speicher befindlichen s_{t-1} , detektiert wird, wird ein neuer Zyklus gestartet.

Im ersten Schritt wird, wie es in Abbildung 47 dargestellt ist, die aktuelle Situation s_{t_0} , das sind die elektrischen Schaltzustände aller Geräte in der Wohnung (δ_t) sowie die Umweltbedingungen (ω_t), sowohl durch den Anwender (mit seinen Sinnesorganen) sowie durch ENKOS mithilfe entsprechender Sensorik erfasst und in verarbeitbare Daten umgewandelt.

$$s_{t0} = \begin{bmatrix} \delta_t \\ \omega_t \end{bmatrix} \quad (65)$$

Auf diese Situation reagiert der Anwender seiner Erfahrungen und Wünsche je nach Präferenz, das Resultat ist die Sollaktion a'_{t1} . Diese Aktion ist nicht in Abbildung 47 dargestellt, da sie wiederum eindeutig zur Sollaktion s'_{t1} führt. Damit stellt sie die tatsächliche Steuerung der Verbraucher dar und ist gleichzeitig die Führungsgröße (für das Teilziel) von ENKOS, das später im Rahmen der Komfortsteigerung die Entscheidungen des Anwenders adaptieren soll.

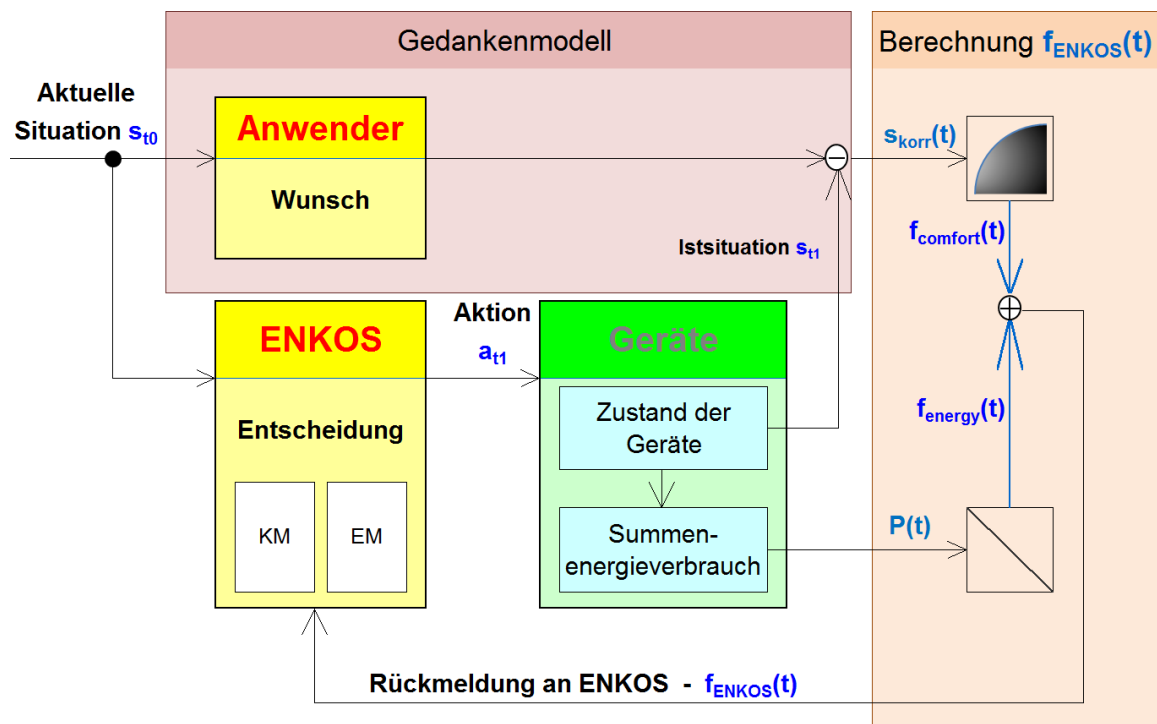


Abbildung 47. Systemtopologie (während der Anwendungsphase, KM = Komfortmaximierung, EM = Energieminimierung).

Parallel dazu, nachdem die Aktion a_{t1} ausgeführt wurde und sich die Situation s_{t1} einstellt, wird die Zielfunktion (vgl. (63)) berechnet. Für einen der

beiden Teilziele wird dazu die gesamte elektrische Leistungsaufnahme gemessen und daraus die elektrisch benötigte Energie $E(t)$ für einen definierten Zeitbereich berechnet.

Für das zweite Teilziel, die Komfortmaximierung, wird prinzipiell zwischen der Lern- und der Anwendungsphase unterschieden.

Lernphase

Während der Lernphase, wenn die Wissensbasis noch nicht hinreichend die Umwelt repräsentiert, berechnet sich $s_{\text{kor}}(t) = 1$, da per Definition der Anwender selbst ENKOS belehrt und noch gar keine Vorhersage durchgeführt wird. Damit ergibt sich per se die maximale Nutzerzufriedenheit, deshalb ist eine separate Auswertung an dieser Stelle nicht nötig.

Anwendungsphase

Wenn die Prognosesicherheit, und damit der Modellreifegrad, groß genug ist²³, übernimmt ENKOS die Steuerung der Verbraucher mittels Bildung von a_{t1} , und der Anwender bestraft bzw. belohnt diese Entscheidungen. Damit befindet sich das System in der *Anwendungsphase*. Im Unterschied zur ersten Phase wird dem Anwender und ENKOS zwar ebenfalls eine bestimmte Situation s_{i0} präsentiert (s. Abbildung 47), doch die auszuführende Entscheidung trifft nun ENKOS. Als Resultat wird die Aktion a_{t1} den Zustand der Geräte ändern und nach Realisierung der Aktion die neue Situation s_{t1}

²³ Diese Beurteilung erfolgt durch die Überschreitung einer bestimmten Belohnung (also Vorhersagesicherheit), wie später dargestellt wird.

erzeugen. Parallel dazu wünscht sich der Anwender selbst ebenfalls (gedanklich) eine neue Situation s'_H (Sollsituation). Diese beiden Situationen werden im nächsten Schritt gedanklich miteinander verglichen (indem bestimmte Geräte ggf. durch den Nutzer in einer bestimmten Zeit korrigiert werden). Diese Werte werden in $s_{\text{kor}} (vgl. (52))$ umgerechnet und der Wert fließt direkt in die Zielfunktion (s. dazu (63) und (64)) ein.

Als zweite Größe, die für die Zielfunktion benötigt wird, wird auch hier die gesamte elektrische Leistungsaufnahme gemessen und in den aktuellen Energiebedarf $E(t)$ umgerechnet. Anschließend ergibt sich f_{ENKOS} für den aktuellen Zeitpunkt als Bewertungsgröße für ENKOS.

Im Wesentlichen soll die Handlungsentscheidung, die ENKOS trifft, sowohl vom menschlichen Verhalten als auch vom Summenenergieverbrauch aller elektrischen Verbraucher abhängen, da nur dann die Maximierung der Zielfunktion möglich ist. Außerdem soll diese Komponente sich anhand der Antwort von der Umwelt, also dem Wert der Zielfunktion, adaptiv entwickeln können, um eine Maximierung der Zielfunktion anzustreben.

6.2 Systemintegration zum „Kognitiven Learning Classifier System“ (KLCS)

Die LCS mit integrierten analytischen Methoden zur Energieminimierung werden nachfolgend als „Kognitive Learning Classifier Systeme“ (KLCS) bezeichnet, da in der Literatur bislang kein vergleichbarer Ansatz gefunden wurde. Als Erweiterung zu „klassischen“ LCS, die mit einem bestimmten Reifegrad (wenn der durchschnittliche Zielfunktionswert einer Generation von Regeln über einem bestimmten Schwellwert ist) mit Hilfe derer Regelbasis ein **Verhaltensmodell** der sie bestrafenden oder belohnenden Umwelt

(also der Personen im Haushalt) in bestimmten, wesentlichen Arbeitsebenen (Building Blocks) abbilden, enthalten die KLCS zusätzlich ein **physikalisch-logisches Modell** desselben Haushalts als Grundlage für die drei vorgestellten Methoden zur Energieminimierung, die dieses Modell nutzen. Bei der Integration beider Ansätze zu einem gemeinsamen sind werden grundsätzliche Varianten diskutiert:

- die **Einschränkung des Definitionsbereichs** für den Zustandsraum,
- die Integration weiterer Agenten als „**virtuelle Nutzer**“ sowie
- den zyklischen **Aufruf eines externen Lehrers**.

Teilziel	Modell-ansatz	Einschrän- kung Defi- ni- tionsbe- reich	Virtueller Nutzer	Aufruf ex- terner Leh- rer
Komfortmaxi- mierung	LCS/ Ver- haltensmo- dell	Möglich	Möglich	Möglich
Energiemini- mierung/ Nichtwahr- nehmbarkeit	Logisches Modell	Möglich	Möglich, aber nicht konsequent umgesetzt	Keine kon- sequente Umsetzung möglich
Energiemini- mierung/ Sub- stitution und Widerspruchs- unterdrückung	Physikali- sches Mo- dell	Möglich	(perma- nente er- zwungene Kompro- missbil- dung)	

Tabelle 11. Gegenüberstellung der Integrationsmöglichkeiten der drei Modellansätze in ein Gesamtmodell.

Dabei sollen alle drei Topologien vorgestellt und miteinander verglichen werden. Daraus wird die Integration aller Methoden zu einem Gesamtmodell definiert.

In Tabelle 11 werden deshalb mehrere mögliche Varianten hinsichtlich ihrer Vor- und Nachteile diskutiert, um daraufhin einen geeigneten Ansatz als Grundlage für die KLCS auszuwählen. Dabei zeigt sich, dass die Integrationsmethode des eingeschränkten Definitionsbereichs des Aktionsraums die geeignete ist. Diese wird nachfolgend weiter entwickelt.

6.2.1 Einschränkung des Definitionsbereichs

Bei dieser Integrationsmethode wird bei jeder Mutation bzw. sonstigen Veränderung des Aktionsteils einer Regel mit Hilfe des Logischen sowie des Physikalischen Modells darauf geachtet, dass nur gültige Zustände existieren können. Sobald eine Situation generiert wird, die in der Natur so nicht vorkommen kann, wird diese entsprechend zu einer gültigen verändert. Wenn beispielsweise eine Regel Ψ_v generiert wurde, die das Einschalten einer Lampe in einem Zimmer verlangt, während dort keine Person anwesend ist, wird diese Regel entsprechend geändert, bevor sie zur Anwendung kommen darf. Dieser Fall ist weiter oben bereits vorausgesetzt worden.

Der Vorteil dieses Ansatzes ist, dass er alle Informationen benutzt, die auch erhoben werden, da alle Modelle ihre Berücksichtigung finden. Außerdem können alle Anforderungen an das System ENKOS abgebildet werden.

Nachteilig ist jedoch, dass unbekannte Zustände nie durchsucht werden. Das bedeutet, wenn die Kombination aus Verhaltensmodell und logischem Modell im Moment eine ungültige Kombination darstellen, weil der Weg noch

nicht gelernt wurde, wird er automatisch „korrigiert“. Deshalb ist die Lernfähigkeit hier gestört.

Der geeignete Zeitpunkt einer Modifikation einer Regel ist, wann immer die Regel geändert wird. Das ist der Fall, wenn die Genetischen Algorithmen zum Einsatz kommen.

6.2.2 Agenten dienen als virtuelle Nutzer

Bei diesem Ansatz wird die Sollsituation nicht allein durch das menschliche Verhalten bestimmt, sondern durch die permanente Konkurrenz zwischen dem Anwender und einem bzw. mehreren weiteren, virtueller Nutzer, die als „Energiminimierer“ koexistieren und Regeln für die Reduzierung der elektrischen Leistungsaufnahme erstellen. In Abbildung 48 ist die Topologie dieses Ansatzes gezeigt.

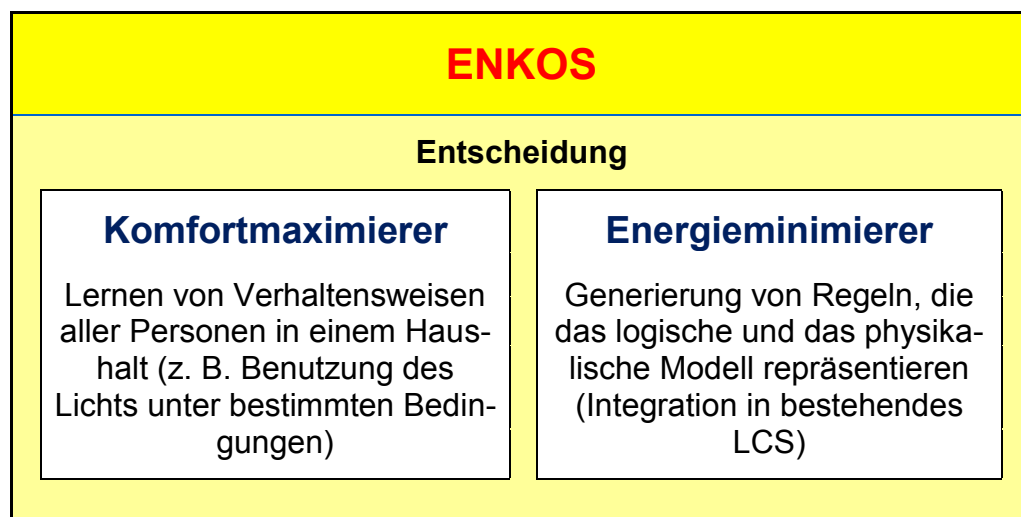


Abbildung 48. Integration eines virtuellen Nutzers, der die Methoden der Energie-minimierung widerspiegelt. Der Übersichtlichkeit halber ist nur der Block „ENKOS“ aus Abbildung 47 dargestellt, da sich das restliche System nicht verändert.

Der Vorteil dieser Topologie ist, dass alle Methoden gleichberechtigt koexistieren und gemeinsam den Sollzustand generieren. Daraus ergibt sich

aber der wesentliche Nachteil, dass durch diese Kompromissbildung völlig unsinnige Zustände möglich sind. Auch werden durch die ständigen Kompromisse die einzelnen Modelle selten konsequent zur Anwendung kommen. Das führt dazu, dass Zwischenzustände generiert werden, die den Nutzer nicht gefallen können und damit permanent bestraft würden.

6.2.3 Aufruf eines externen Lehrers

Bei diesem Ansatz handelt es sich um einen externen Lehrer, der omnipotent alle Regeln während deren Anwendung überwacht und bei Bedarf im Sinne der Energieminimierung „korrigiert“. Damit ist zwar der Vorteil, dass ständig gültige Regeln im Modell vorhanden sind, gegeben. Allerdings steht dem der Nachteil gegenüber, dass damit ein nicht wirklich integrierter Ansatz geschaffen wird. Das bedeutet, dass zwei Entscheidungsebenen existieren, die Anwendungs- und die Lehrerebene. Desweiteren ist dieser Ansatz keine wesentliche Verbesserung der integrierten Variante „Einschränkung des Definitionsbereichs“, der bereits eine Integration darstellt. Aus diesem Grund wird hier nicht weiter auf diesen Ansatz eingegangen.

Konsequent der Methode der Abnehmenden Abstraktion folgend, wird damit ein Teil der Umwelt als sehr spezielles (analytisches) White-Box Modell gebildet, während ein anderer Teil auf einer abstrakteren, regelbasierten Ebene als Grey-Box modellierbar ist. Bei der methodischen Integration der beiden gibt es verschiedene Ansätze, die zunächst vorgestellt und anschließend diskutiert werden.

Als erstes gibt es die Möglichkeit, dass beide prinzipiellen Lösungsansätze gleichberechtigt miteinander um die beste Lösung konkurrieren, indem sowohl die Regeln als auch die analytischen Methoden eine Entscheidung tref-

fen und anschließend abgleichen. Bei diesem Ansatz wird jedoch der wesentliche Denkansatz außer Acht gelassen, dass das Raummodell ein harter Fakt ist, der nicht mit einem unschärferen Verhaltensmodell konkurrieren sollte, damit keine Informationen künstlich verwechselt werden müssen. Da die zu findende Lösung stabil sein muss, ist auch selten ein reiner Kompromiss möglich, sondern vielmehr die Anwendung des einen oder des anderen Vorschlags.

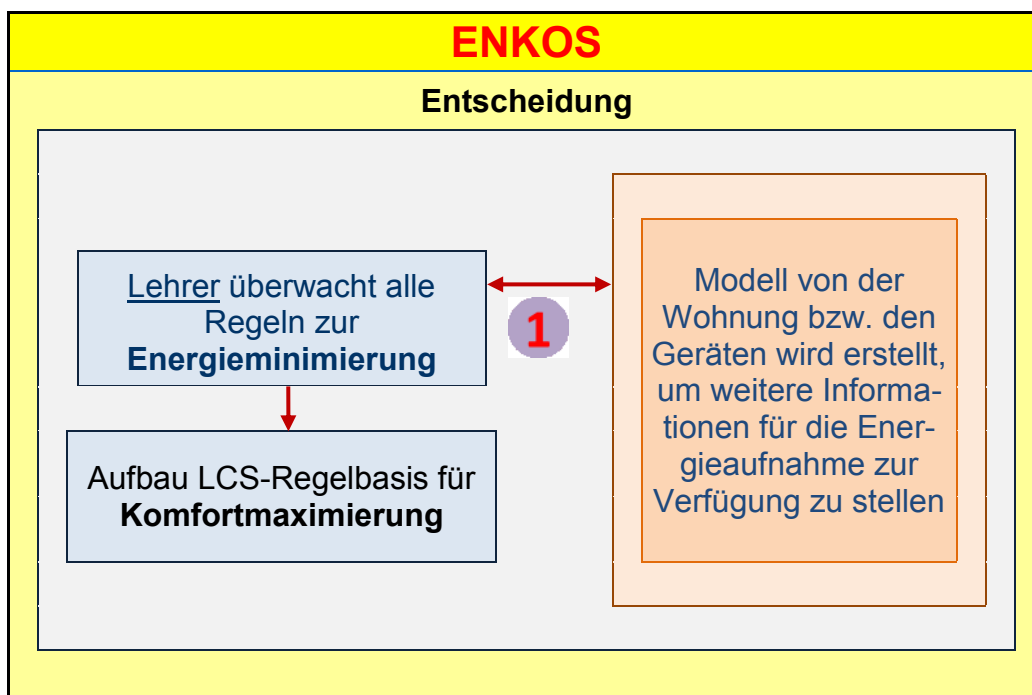


Abbildung 49. Externer Lehrer zur Überwachung der Regelbasis. Zur Vereinfachung ist nur die Komponente ENKOS dargestellt, die restliche Systemtopologie bleibt gleich.

Deshalb wird der Ansatz weiterentwickelt zu einem Master-Slave-Modell, bei dem der LCS-Ansatz das wesentliche Gerüst von ENKOS bildet. Dabei werden die Regeln infolge der Rückmeldung des Verbrauchers gebildet und modifiziert, wie es beim LCS üblich ist, bevor sie mit Hilfe des Raummodells mit Hilfe der vorgestellten Methoden angepasst werden. Während der Prozess in Prozess 1 (die Kognitiven Methoden im Schritt 7 in Abbildung

54) gezeigt ist, soll die Abbildung 50 illustrieren, wie eine Regel dabei konkret verändert wird. Der Zusammenhang zu Abbildung 49 ist durch die „1“ im Kreis dargestellt. An dieser Stelle benutzt der „Lehrer“ verfügbare Informationen über die Geräte in der Wohnung sowie die Wohnung selbst, um die Regeln effektiver in Bezug auf die Energieminimierung zu gestalten.

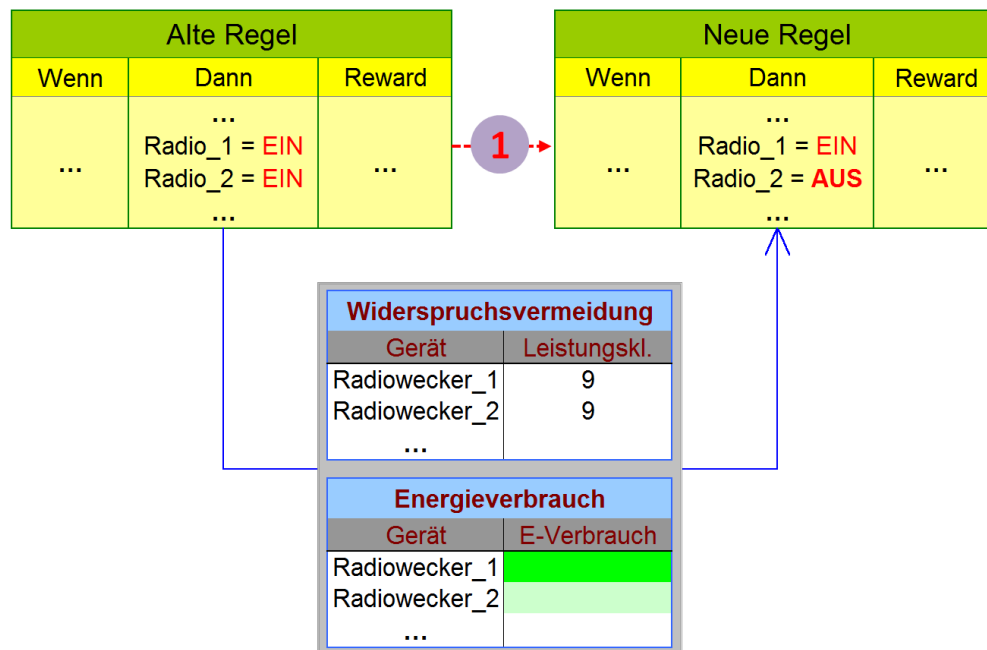


Abbildung 50. Beispiel einer Modifikation einer Regel infolge der Anwendung der Kognitiven Methoden „Widerspruchsvermeidung“ sowie „Substitution“ (zur Ermittlung des einzuschaltenden Geräts). Der logische Pfad „1“ ist der Pfad, der in Abbildung 49 dargestellt ist.

In diesem Beispiel entsteht in der Regel infolge des aktuellen Lernschritts (z. B. durch Anwendung der Mutation) ein Widerspruch, da sowohl Radiowecker 1 als auch Radiowecker 2 eingeschaltet werden sollen, die beide in der gleichen Dienstleistungsklasse enthalten sind und deren Einschalten sich demzufolge widersprechen würde. Um zu entscheiden, welches Gerät weiterhin eingeschaltet bleiben soll, wird im aufgebauten Modell der Wohnung, in dem auch die Energieverbräuche aller Verbraucher gespeichert sind, das

Gerät mit dem kleineren Energieverbrauch ermittelt und eingeschaltet. Im vorliegenden Fall ist das Radiowecker_1, der mit grün in der unteren Tabelle markiert ist, während der in Bezug auf die Stromaufnahme „verschwendereichere“ Radiowecker_2 abgeschaltet wird. Damit ist der Konflikt gelöst. Im nächsten Schritt sollen alle verwendeten Methoden der KLCS wiederum an der wirklich vorhandenen Umwelt gespiegelt werden. Damit soll der Ansatz in Bezug auf die gegebene Aufgabenstellung, menschliche Verhaltensweisen abzubilden, verifiziert werden.

6.2.4 Einbindung der analytischen Komponenten zur Energieminimierung

Die Einbindung der analytischen Komponenten erfolgt über die Regelbasis direkt. Das heißt, jede Regel wird im Rahmen der Verdichtung permanent auf deren Konsistenz geprüft, indem sie mit dem vorhandenen Raummodell und dem Gruppenmodell abgeglichen und ggf. korrigiert wird. Also kann niemals eine Regel zur Anwendung kommen, deren Aktion gegen einen vorhandenen Modellstand verstößt.

6.2.5 Generalisierung als menschliche Abstraktion

Die Generalisierung von LCS kann als menschliche Fähigkeit zur Abstraktion (induktives Schließen) in diesem Anwendungsfall interpretiert werden. Bei der Lichtstärke und der Uhrzeit haben die geringen Generalisierungsraten keinen merklichen Erfolg gezeigt, sodass dort weitere Parameter ins Modell eingeflossen sind.

6.3 Abbildung menschlichen Verhaltens mit Verhaltens-erwerbsmethoden

Im nächsten Schritt wird zum einen das verhaltenspsychologische Modell mit verschiedenen, in Frage kommenden Methoden des maschinellen Lernens verglichen, um daraus ein adaptiv lernendes System zu entwickeln. Dabei werden in den nächsten Abschnitten Neuronale Netze und Learning Classifier Systeme als die beiden möglichen Methoden implementiert und näher untersucht.

6.3.1 Lösungsansatz Neuronale Netze

Zunächst wurde zur technischen Adaptierung des Verbraucherverhaltens ein Neuronales Netz implementiert, das Muster aus präsentierten Beispielen lernt und klassifiziert. Hintergrund ist, dass Neuronale Netze sehr gut hochgradig nichtlineare Zusammenhänge widerspiegeln. Im ersten Schritt wurde dabei eine feste Netzstruktur gewählt, die sich allerdings nach wenigen Versuchen bereits als zu starr erwies. Denn je nach Komplexität der Aufgabe wurden schnell Undersampling oder Overfitting sichtbar.

Deshalb wurde im nächsten Schritt für alle Parameter eines Neuronalen Netzes (vollständig vorwärtsverkettet, Adaptierung der Gewichte mit Backpropagation-Lernalgorithmus) ein Wertebereich definiert und jede mögliche Kombination dieser Parameter ergibt eine eigene Netzkonfiguration. Diese wurde mit einem Standardlern Datensatz belehrt und mit einem Standarddatensatz verifiziert. Alle Parameter sind in Abbildung 51 gezeigt, wobei das dort eingestellte Netz mit genau zwei Hiddenschichten gebildet wird, da für alle anderen Hiddenschichten die Wertebereiche „0“ sind. Die Anzahl der Eingangsschichten reicht im gezeigten Beispiel von 2 bis 3, sodass hier eine

Parameteridentifikation (Feature Selection) stattfinden kann. Die Lernrate ist in diesem Beispiel bei 0.5 festgeschrieben, während kein Momentumterm vorhanden ist. Es werden allerdings beide Lernraten, eine konstante und eine variable ausprobiert, was als $\text{Typ}_{\text{Lernrate}}$ definiert ist.

		Linearer Anstieg (Arbeitsbereich f. 100 %-Suche)			Toleranz- bereich (EvoAlg)		
		Start	Inkr.	Ziel	MIN	MAX	
Netztopologie	# Input-Knoten	2	1	3	2	3	
	Hiddenschichten	# Hid ₀	2	1	2	1	4
		# Hid ₁	2	1	2	1	4
		# Hid ₂	2	1	2	1	4
		# Hid ₃	0	1	0	0	4
		# Hid ₄	0	1	0	0	0
		# Hid ₅	0	1	0	0	0
		# Hid ₆	0	1	0	0	0
		# Hid ₇	0	1	0	0	0
		# Hid ₈	0	1	0	0	0
	# Hid ₉	0	1	0	0	0	
# Output-Kn.	1	1	1	1	1		
	Lernrate	0,5	0,2	0,5	0,1	0,9	
	Momentumterm	0	0,01	0	0	0	
	# Muster	4			0	10	
Max. Fehler/ Muster		0,1			[abs. Wert]		
Max. # Lernversuche		10000			[abs. Wert]		
Aktivierungsfunktion		1	1	1	1	1	
Typ _{Lernrate}		1	1	2	1	2	
Simulation starten							

Abbildung 51. Definition der Wertebereiche aller Parameter für alle möglichen Netzkonfigurationen.

Das Bewertungskriterium, wie gut eine bestimmte Netzkonfiguration des Verbraucherverhaltens vorhersagen kann, ist die Zielfunktion, die weiter oben definiert wurde und in (63) dargestellt ist. Während in der Netzantwort per Vorwärtspropagierung eine Reaktion auf eine gegebene Umweltsituation

gebildet wurde, wird diese anschließend mit dem Sollzustand verglichen und der Fehler in die Komfortabweichung umgerechnet. Außerdem wird die Energieaufnahme berechnet und alles in der Zielfunktion dargestellt.

In Abbildung 52 wurden sieben aufeinanderfolgende Generationen, die durch Evolutionäre Algorithmen gebildet wurden, berechnet. Dabei wurde aus jeder Generation das Netzwerk, das als erstes einen bestimmten Reifegrad des menschlichen Verbraucherverhaltens am schnellsten abbildet, als Grundlage für die nächste Generation benutzt. Das bedeutet, dass diejenige Topologie, die am erfolgreichsten ist, sich weiterentwickelt und die anderen ignoriert werden. Es zeigt sich erstens, dass sehr viele Lernbeispiele präsentiert werden mussten, bis das Neuronale Netzwerk in der Vorwärtspropagierung das Verbraucherverhalten adaptierte und zweitens, dass von vornherein bessere Topologien auch bessere Nachkommen erzeugen. Die Kurve „MIN“ in Abbildung 52 spiegelt in jeder Generation dabei das Neuronale Netzwerk wieder, das am schnellsten das Nutzerverhalten adaptiert, „MAX“ das jeweils schlechteste und „MEAN“ dasjenige mit der durchschnittlichen Anzahl an Beispielen, bis das Netzwerk einen gewissen Reifegrad erreicht.

Da zum einen die Anzahl der Netze sehr schnell explodiert, um sehr unterschiedliche Arbeitspunkte des Verbraucherverhaltens abzubilden, und selbst mit mehreren Rechnern nicht mehr berechenbar ist und zum anderen auch augenscheinlich schlechte Parameterkombinationen durch die vollfaktorielle Versuchsplanung ausprobiert wurden, wurde im dritten Schritt diese Parametrierung durch Evolutionäre Algorithmen vorgenommen. Damit wird die Netzkonfiguration zielgerichtet weiterentwickelt, basierend auf den Erfahrungen bisher ausprobierteter Netzkonfigurationen. Die zulässigen Grenzen dafür sind in Abbildung 51 auf der rechten Seite gezeigt.

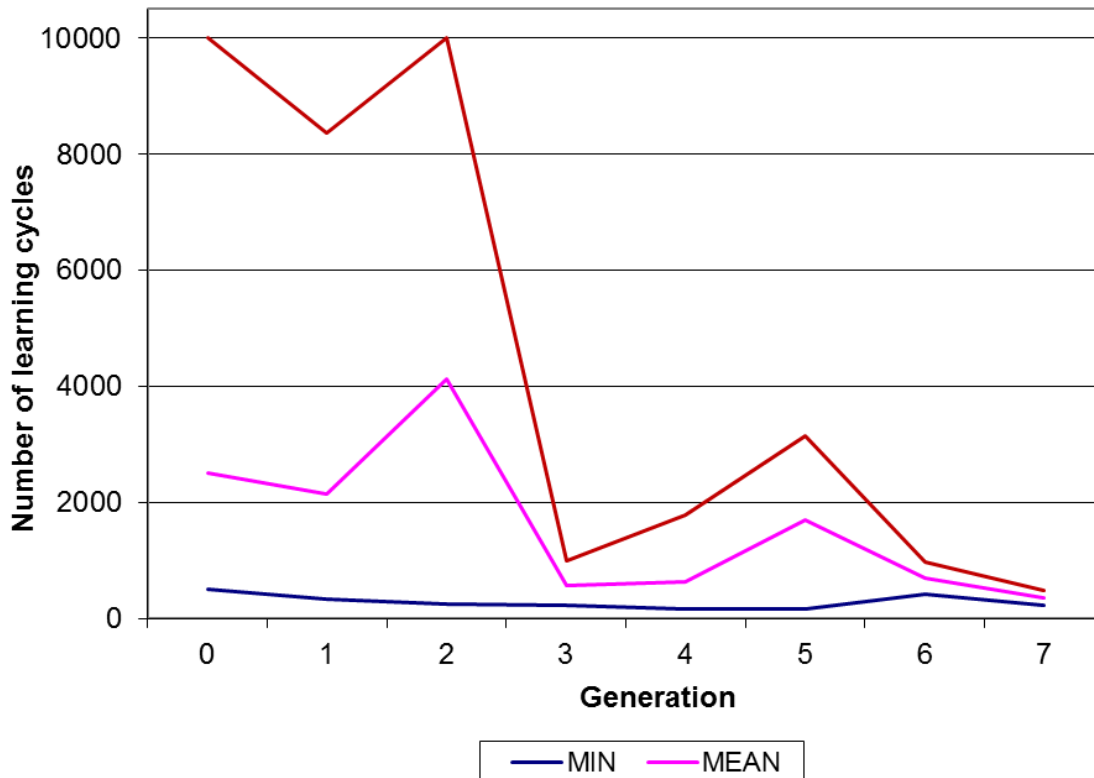


Abbildung 52. Anzahl der Lernzyklen, bis ein definierter Fehlerwert unterschritten wird. Dabei wurden sieben Generationen von Neuronalen Netzwerktopologien (per Evolutionäre Algorithmen) simuliert und in jeder Generation fünf Neuronale Netzwerke.

In Abbildung 53 sind mehrere Generationen von Netzen abgebildet, wobei das Vergleichskriterium hierbei die Anzahl der Lernschritte ist, bis der Wert der Zielfunktion f_{ENKOS} einen bestimmten Schwellwert überschreitet²⁴. Dabei ist sichtbar, dass bestimmte Netztopologien die Adaption des Verbraucherverhaltens deutlich besser (d. h. schneller) meistern als andere, und sich von Generation zu Generation eine Verbesserung bis hin zur besten Lösung einstellt, die mit Neuronalen Netzen (dieser Topologie) möglich ist.

²⁴ Damit ist eine bestimmte Systemreife erreicht.

		#INP	#HID ₀	#HID ₁	#HID ₂	#OUT	LRate	Typ _{Lernrate}	ERG
CHILD	5_19	2	3	3	3	1	0,71	Const.	330
CHILD	9_16	3	1	2	1	1	0,63	Const.	617
CHILD	9_17	3	1	2	1	1	0,48	Const.	772
CHILD	5_18	2	3	2	3	1	0,53	Const.	984
CHILD	9_15	3	1	3	1	1	0,23	Const.	1430
CHILD	5_20	2	3	1	5	1	0,18	Const.	1911
CHILD	2_21	2	4	5	1	1	0,35	Const.	2697
CHILD	2_22	4	3	3	1	1	0,30	Const.	8361
CHILD	5_19_23_28	2	3	3	3	1	1,12	Const.	216
CHILD	5_19_23_29	2	3	3	3	1	0,85	Const.	265
CHILD	5_19_23_30	2	3	3	3	1	0,64	Const.	794
CHILD	5_19_23_27	2	3	4	3	1	0,69	Const.	986
CHILD	5_19_23_28_33	2	3	3	3	1	1,53	Const.	165
CHILD	5_19_23_28_32	2	3	3	3	1	0,99	Const.	259
CHILD	5_19_23_28_34	2	3	2	4	1	1,23	Const.	286
CHILD	5_19_23_28_31	2	3	3	3	1	0,87	Const.	1777

Abbildung 53. Vergleich verschiedener, mit Hilfe Evolutionärer Algorithmen gebildeter Netzkonfigurationen.

In Abbildung 53 ist ein Simulationslauf dargestellt, in der jede Zeile einer Simulation mit den gleichen Eingangsdaten (standardisiertes Verbraucherverhalten) entspricht. Die Spalte „#INP“ ist die Anzahl der Eingangsneuronen, wobei hier eine Parameteridentifikation erfolgt, da sich ab der dritten Generation (in der zweiten Spalte sind mindestens drei Zahlen für die Topologie vorhanden) nur noch Neuronale Netze mit zwei Eingangsgrößen durchsetzen. Die Schichten „#HID_x“ stehen, von links beginnend, für die Neuronen in der Hiddenschicht *X* von der Eingangs- bis zur Ausgangsschicht. Die Lernrate wurde hier per Evolutionärem Algorithmus modifiziert und ergibt bei 1,62 das beste Ergebnis (also eine definierte Abbildung des Verbraucherverhaltens nach 154 Lernschritten), das in der sechsten Generation erzielt wurde.

Nicht mehr dargestellt ist, dass sich das Ergebnis in weiteren Generationen nicht mehr überbieten ließ und deshalb die Simulation wenig später automatisch abgebrochen wurde. Die Durchführung mit allen möglichen Kombinationen hätte, um dieses Ergebnis zu erreichen, mehrerer hundert Simulationen

nen bedurft. Deshalb hat sich der Ansatz, die Netztopologien mit Hilfe Evolutionärer Algorithmen gezielt zu generieren, als wegweisend erwiesen.

Im Ergebnis war das Neuronale Netz unfähig, sich auf verändertes Verhalten für die gegebene Aufgabenstellung einzustellen. Das geäußert sich darin, dass für einen Vektor von Mustern ein optimales Netz gebildet wird, das für eine kleine Änderung von Eingangsdaten wiederum deutlich schlechter angepasst ist (also viel längere Lernzeit benötigt, um einen bestimmten Reifegrad zu erreichen). Das ist dadurch zu erklären, dass das Neuronale Netz an sich eine intrinsische Generalisierungsfähigkeit, aber keine ausgeprägte Abstraktionsfähigkeit besitzt. Desweiteren adaptiert es das menschliche Verhalten nicht per se, sondern vielmehr die Eingangsdaten, die ihm präsentiert werden. Damit ist die Wissensverdichtung nach der Building Blocks-Hypothese nicht erkennbar.

Ein weiterer, entscheidender Nachteil Neuronaler Netze ist, dass die gelernten Abhängigkeiten nur sehr schwierig auslesbar sind. Aufgrund deren Hypernichtlinearität muss das Netzwerk mit allen möglichen Eingangsdaten beaufschlagt und die Netzantwort dabei ausgewertet werden (eine Methode des Reverse Engineering). Außerdem ist eine große Menge an Daten zur Belehrung nötig. Die Ermittlung des maximal erreichbaren Werts für f_{ENKOS} hat sich als sehr langwierig herausgestellt. Als Vergleichskriterium ist deshalb diese Zeitdauer geeignet, da sie direkt proportional zum maximal erreichbaren Reifegrad der Modellstruktur ist.

Deshalb eignen sich Neuronale Netze nicht für die Lösung der gegebenen Problemstellung, wenngleich sie in der Lage sind, hypernichtlineare Abhängigkeiten als Grundlage menschlichen Verhaltens zu lernen. Sobald jedoch unbekannte Arbeitsbereiche erschlossen werden, die anderen Gesetzmäßigkeiten (= menschlichen Gewohnheiten) gehorchen, scheitern alle gewählten

Topologien rasch. Dazu kommt, dass im Gegensatz zu einer intrinsischen Korrektur, in der beispielsweise allmählich Gewichte zu null werden (also Kanten verschwinden) oder sich bilden, eine Änderung von außen zu einer deutlichen Änderung der Ausgangswerte bei gleich bleibenden Gewichten führt.

Mit Hilfe der Neuronalen Netze konnte der systematische Aufbau sowie die Richtigkeit der Umwelt abgebildet werden, auch die relevanten physikalischen Größen (Umweltbedingungen, Verbraucherverhalten, elektrische Geräte) wurden abgebildet. Die Wirtschaftlichkeit ist mit diesem Ansatz nicht darstellbar, da die Simulationszeit teilweise mehrere Tage überstieg und unreife Abbildungen der Umwelt keine ausreichende Nutzerakzeptanz bieten. Desweiteren ist auch die Vergleichbarkeit verschiedener Ansätze nicht ausreichend gegeben, da sich mit jeder Änderung der Netztopologie völlig andere Abhängigkeiten zwischen den Knoten ergeben. Damit ist jedes Mal ein neuer, vollständiger Lernvorgang nötig.

Der wesentlichste Nachteil dieses Ansatzes bestand jedoch darin, dass die Klarheit nach den Forderungen in Abschnitt 2.2 nicht gegeben war. Die gespeicherten Informationen sind in der Netztopologie sowie den Gewichten zwischen den Neuronen kodiert und keineswegs direkt auslesbar. Deshalb ist die Gefahr sehr groß, dass an den Grenzen des belehrten Arbeitsbereichs durch die gegebenen Beispiele unsinnige Handlungen durch ENKOS vollzogen würden, die sofort die Akzeptanz des gesamten Ansatzes infrage stellen würden. Aus diesem Grund wird im nächsten Abschnitt ein teilweise darauf aufbauendes Konzept entwickelt.

6.3.2 Modellierung mit Hilfe von Learning Classifier Systemen

6.3.2.1 Modellierung mit Hilfe von Learning Classifier Systemen

Als zweite Methode zur Adaptierung menschlicher Entscheidungen in einer bestimmten Situation wurden Learning Classifier Systems als klassischen Vertreter der Verhaltenserwerbsmethoden implementiert. Hierbei werden die Vorteile der Neuronalen Netze, die Abbildung von multidimensionalen und hypernichtlinearen Zusammenhängen sowie die Generalisierungsfähigkeit aufgegriffen und durch die Fähigkeit der Building Blocks erweitert. Gleichzeitig verschwindet der schwer durchschaubare Charakter der Black Box-Methode, indem die Wissensbasis transparent aufgebaut ist und die Methode kommt dem Vorgehen von Menschen während deren Entscheidungsfindung noch näher [72].

Kernstück des hier implementierten ist die Wissensbasis (Population $[P]$), die in der vorliegenden Aufgabe das Anwenderverhalten im Haushalt abbildet [142]. Je nach Lernfortschritt hat dieses Modell einen gewissen Reifegrad, der die Gewohnheiten des Nutzers bis zu einer gewissen Tiefe abbildet. Die Wissensbasis besteht aus maximal $||\Psi||$ Regeln Ψ , die einer Situation $\Psi.s_{t_0}$ eine mögliche Aktion $\Psi.a_{t_1}$ zuordnen. Die Indizes t_0 und t_1 bedeuten dabei, dass die Situation zum aktuell gelesenen Zeitschritt t_0 erfasst wird, während sich die Aktion im nächsten Zeitschritt t_1 durchgeführt wird.

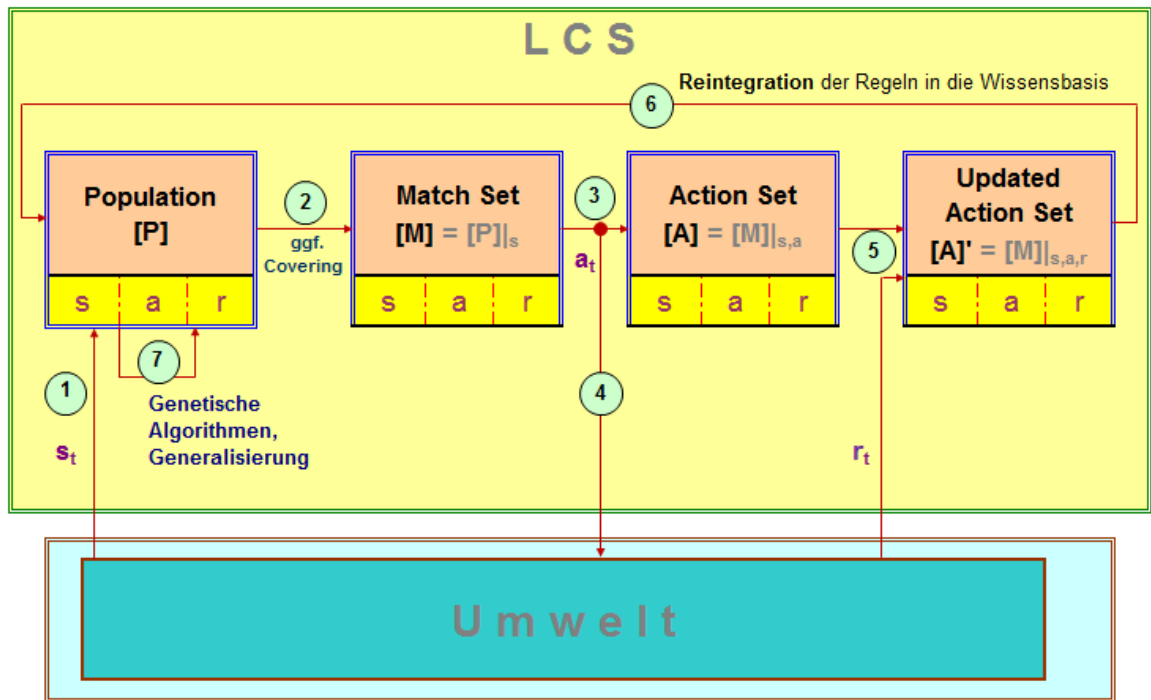


Abbildung 54. Topologie des LCS für die konkrete Problemstellung. Die Umwelt besteht aus einem „steuerbaren“ (z. B. elektrische Geräte) und einem „nicht steuerbaren“ Teil (z. B. Temperatur, Helligkeit).

LCS werden im nächsten Schritt auf die konkrete Problemstellung angepasst, indem einerseits die Schnittstellen zur Umwelt und andererseits die interne Struktur angepasst werden. Die Topologie eines LCS sowie die Schnittstelle zur Umwelt ist prinzipiell in Abbildung 54 dargestellt. Daraus ergibt sich die konkrete Grundstruktur von ENKOS in Abbildung 55.

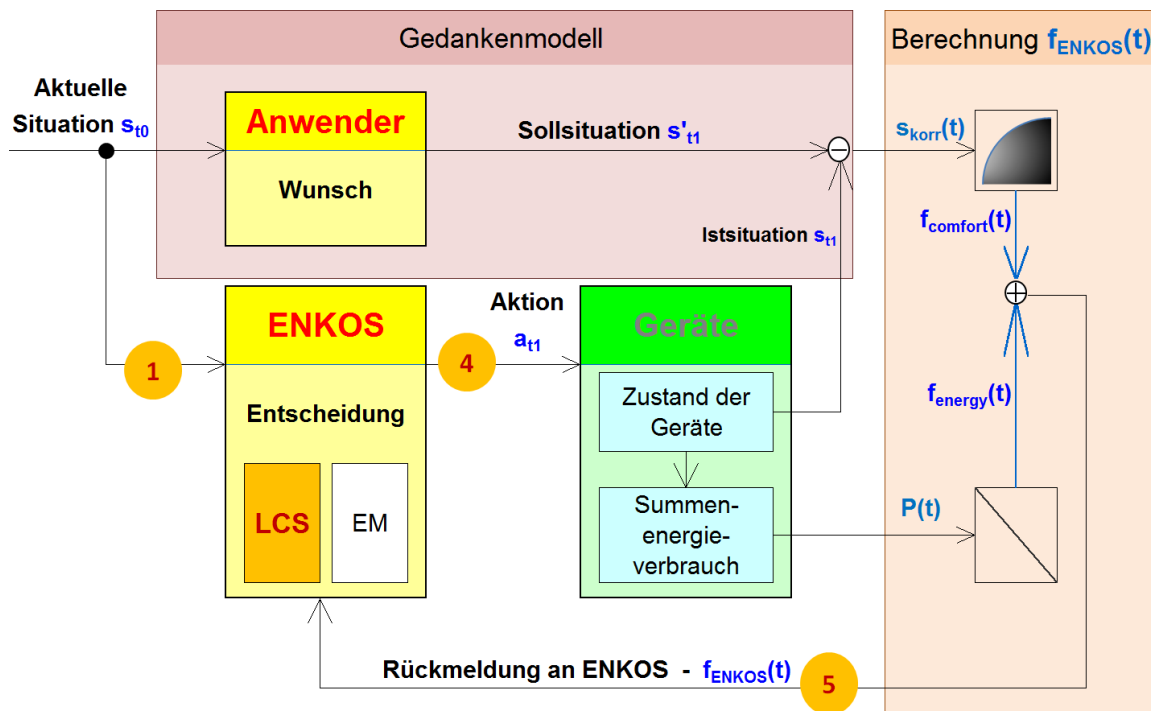


Abbildung 55. Integration der LCS in ENKOS. Die Schritte 1, 4 und 5 in den gelben Kreisen entsprechen den Schritten in Abbildung 54. Damit ist die Schnittstelle zur Umwelt dargestellt (EM = Energieminimierung).

Für die Initialisierung der Wissensbasis gibt es zwei Ansätze. Entweder ist eine Population vollständig mit Jokern im Situationsteil gefüllt, während bei den Aktionen zufällig Handlungsanweisungen zusammengestellt werden. Das ist sinnvoll für Optimierungsprobleme, bei denen die Aktion keinen physikalisch durchzuführenden Prozessschritt darstellt, sondern „nur“ Rechenoperationen und der Zustandsraum schnell adaptiert werden muss. Das vorliegende System hingegen startet mit einer leeren Wissensbasis, was der zweiten Möglichkeit entspricht. Hintergrund ist, dass im Haushalt so wenige Fehlentscheidungen wie möglich stattfinden sollen, um den Anwender nicht unnötig zu enttäuschen²⁵. Damit ist ENKOS zuerst lediglich Beobachter, bis dessen Wissensbasis einen gewissen Reifegrad erreicht hat. Desweiteren ist

²⁵ Die Enttäuschung drückt sich direkt im Verbraucherkomfort aus und somit im Bewertungskriterium des Systems selbst.

das Nutzerverhalten nach diversen Untersuchungen zu komplex, um ohne zugrunde liegendes Modell zufällig Entscheidungen zu treffen, sowie die jeweiligen Präferenzen des Anwenders am Anfang gar nicht bekannt [128]. Eine Stärke von LCS ist die Parameteridentifikation, die automatisch durch die Vergabe von Jokern durchgeführt wird. Denn wenn sich eine Regel mit vielen Jokern ausbildet, die einen großen Wert des Rewards aus der Umwelt erhält, sind die mit Jokern versehenen Eingangsparameter für die entsprechende Entscheidung nicht wichtig. Beispielsweise könnte sich für das Einschalten des Fernsehers am Abend die Helligkeit als unwesentlich herausstellen. Dann würde beim Eingangsparameter „Helligkeit“ irgendwann ein Joker entstehen, in den Regeln, die sich für das Einschalten des Fernsehers entscheiden.

Im nächsten Schritt werden die Regeln des LCS sukzessive an die Struktur der Umwelt angepasst. Diese bestehen aus der Situation, der Aktion und dem Reward. Diese Schnittstellen sind in Abbildung 55 dargestellt, wo auch die Komponente der „Komfortmaximierung“ durch das LCS ersetzt wurde.

Der Situationsteil der Regeln

Dabei kann eine Situation $\Psi.s_t$ entweder einen bestimmten Zustand der detektierten Umwelt abbilden oder einen Bereich, indem ein Wert zu einem Joker „#“ gesetzt, womit sowohl „0“ als auch „1“ für diesen Zustand repräsentiert wird. Die gelesene Situation besteht aus den Komponenten, die in Tabelle 9 definiert sind. Der Index t bedeutet, dass die Regel zu einem beliebigen Zeitpunkt eingesetzt werden kann.

Im menschlichen Denken entspricht der Einsatz von Jokern der Verallgemeinerung von Wissen (oder dem logischen Schließen). Damit können wir

von den gesammelten Erfahrungen auf zugrunde liegende Mechanismen der Entscheidungsfindung zu schließen. Ein gesetztes Bit „1“ entspricht dem eingeschalteten, ein nicht gesetztes (= „0“) dem ausgeschalteten Zustand eines Geräts. Bei der Lichtstärke sowie der Uhrzeit jedoch sind keine binären Zustände, sondern Bereiche definiert. Das bedeutet, dass Regeln zum Beispiel in einen Bereich der Dämmerung aktiv sind ($\text{Licht}_{\min} = 10\%$ und $\text{Licht}_{\max} = 20\%$), während andere auf den gesamten Tag ($15 \dots 100\%$) bzw. die Nacht ($0 \dots 15\%$) angewandt werden. Die Bereiche sind nicht starr, sodass sie sich auch mit dem Fortschritt der Wissensbasis auf andere Tagesbereiche spezialisieren können.

Der Aktionsteil der Regeln

Die zweite Komponente einer Regel ist die Aktion $\Psi.a_t$. Das bedeutet, dass eine bestimmte Aktion gewählt wird, wenn die dazugehörige Regel eintrifft. Die Aktion ist, entscheidungspsychologisch gesehen, dabei die Absicht, einen neuen Zustand herbeizuführen. Diese umfasst alle steuerbaren Elemente ω_t der Umwelt.

Die Erwartete Belohnung (Reward)

Die dritte Komponente einer Regel ist die zu erwartende Belohnung (Reward) $\Psi.r_t$, wenn man bei einer eingetretenen Situation $\Psi.s$ auch die Aktion $\Psi.a$ auswählt. Diese Belohnung spiegelt damit die Erfolgsaussichten bei Auswahl der entsprechenden Regel wider, wobei dieser Erfolg in ENKOS der Wert der Zielfunktion entspricht. Nachfolgend ist der Zyklus mit EN-

KOS von der Erfassung einer Situation bis zum Update der Regeln beschrieben.

Die Bildung des Match Sets

Nachdem eine Situation s_{t_0} durch die Umweltsensoren zeitdiskret bei jedem Halbstundenintervall detektiert wurde (Schritt 1 in Abbildung 54), werden alle Regeln, deren Situationsteil die detektierte Situation abbilden, in ein Match Set **[M]** (Untermenge der Regelbasis, Schritt 2) übertragen. In der Implementierung in C++ wird dabei das Matching-Bit („isMatchSet“) von „False“ auf „True“ geändert für die entsprechenden Regeln.

Das Covering

Wenn die Wissensbasis für die aktuell detektierte Umweltsituation s_{t_0} keine passende Regel findet, muss eine erzeugt werden, um eine Entscheidung treffen zu können („Covering“). Dabei sind die drei Bestandteile der Regel Ψ_{cov} durch ENKOS zu generieren.

- ➔ die Situation $\Psi_{cov}.s_t$,
- ➔ die Aktion $\Psi_{cov}.a_t$ sowie
- ➔ die zu erwartende Belohnung (Reward) $\Psi_{cov}.r_t$

Die *Situation* der zu erzeugenden Regel $\Psi.s_t$ soll dabei den aktuell detektierten Umweltzustand s_{t_0} abbilden. Das wird dadurch gewährleistet, dass der Zustand zunächst in die Regel kopiert und anschließend generalisiert wird,

indem jedes Bit (im String) mit einer gewissen Wahrscheinlichkeit μ_{Gen} zu einem Joker umgewandelt wird ($\mu_{\text{Mut}} \approx 5\%$).

Für die *Aktion* der gecoverten Regel wird die aktuell ausgeführte Aktion des Anwenders (geführtes Lernen) benutzt ($\Psi.a_t := a_{t1}$). Das bedeutet praktisch, dass die tatsächliche Situation s'_{t1} zum nächsten Zeitpunkt $t = t_1$ in der Simulation gelesen und in die durchzuführende Aktion a_{t1} übertragen wird. Hier ist per Definition die volle Zufriedenheit des Anwenders sichergestellt, da der Anwender selbst die Entscheidung fällt.

Die *Belohnung* $\Psi.r_t$ für diese Regel entspricht dem Funktionswert der Zielfunktion zu diesem Zeitpunkt (vgl. (63)). Im ersten Term berechnet sich $s_{\text{kor}}(t)$ dabei zu 1 (alle Geräte sind korrekt angesteuert worden), womit sich vereinfacht die (66) ergibt.

$$f_{\text{ENKOS,covering}}(t) = 0,8853 + 500 \bullet k_{\text{Norm}} \left(1 - \frac{E(t)}{E_{\text{max}}} \right) \quad (66)$$

Der Wertebereich der Zielfunktion $f_{\text{ENKOS,covering}}(t)$, bei der das Covering einer Regel vorliegt, liegt dabei zwischen 0,8853 (keine Energieersparnis) und 1,7534 (technisch maximal mögliche Energieersparnis).

Danach wird unterschieden, ob die Wissensbasis bereits ihre maximale Größe i_{max} (Anzahl der Regeln) erreicht hat. Ist das nicht der Fall, wird die gecoverte Regel Ψ_{cov} hinzugefügt. Ist die Wissensbasis jedoch bereits vollständig gefüllt, muss eine andere Regel mit der gecoverten überschrieben werden. Dabei gibt es die Möglichkeit, zu einem eine beliebig ausgewählte, zum anderen, eine spezielle Regel zu ersetzen.

Wenn eine beliebige Regel ausgewählt wird, ist die Gefahr groß, dass der Zustandsraum, den das LCS abbildet, verringert wird. Denn es geht Wissen

unkontrolliert verloren, das durch die ersetzte Regel abgebildet wird. Deshalb wird, alternativ zur zufälligen Änderung einer Regel eine gezielt gesucht, die durch die gecoverte ersetzt wird.

Allele					
Lampe 1 Arbeitszimmer	Lampe 2 Arbeitszimmer	PC 1 Arbeitszimmer	PC 2 Arbeitszimmer	Tischlampe Arbeitszimmer	
Situation ξ aus der Population $\Psi.s(\xi)$	1	#	#	1	0
Gelesene Umweltsituation s_{t0}	0	0	1	1	1
Differenz vorhanden?	1	0	0	0	1

Hammingdistanz: **2**

Abbildung 56. Hammingdistanz zwischen zwei Vektoren mit jeweils 5 Elementen (Bits) mit vorhandenen Jokern, von denen jeweils 2 verschiedenen sind.

Um das Verlernen von Wissen zu vermeiden, wird die Hammingdistanz zwischen der Situation der neuen Regel Ψ_{cov} und allen in [P] vorhandenen gebildet. Die Regel mit der kleinsten Hammingdistanz ist dabei die ähnlichste, die auch durch die neue Regel ersetzt wird. Damit ist sichergestellt, analog des menschlichen Lernens, ähnliche bereits vorhandene Kenntnisse in einer Wissensbasis überschrieben werden. In Abbildung 56 und Abbildung 57 sind zwei Beispiele für die Berechnung der Hammingdistanz gezeigt.

Allele					
Lampe 1 Arbeitszimmer	Lampe 2 Arbeitszimmer	PC 1 Arbeitszimmer	PC 2 Arbeitszimmer	Tischlampe Arbeitszimmer	
Situation ξ aus der Population $\Psi.s(\xi)$	1	1	0	1	0
Gelesene Umweltsituation s_{t0}	0	0	1	1	1
Differenz vorhanden?	1	1	1	0	1

Hammingdistanz: **4**

Abbildung 57. Hammingdistanz zwischen zwei Vektoren mit jeweils 5 Elementen (Bits) ohne Joker, von denen jeweils 4 verschieden sind.

Die gecoverte Regel, und nur diese, gehört per Definition als einzige zum Match Set $[M]$, da sie die detektierte Situation genau abbildet.

Auswahl der Aktion a_{t1} :

Als nächster Schritt muss eine entsprechende Aktion a_{t1} aus dem Match Set ausgewählt werden (Schritt 3), mit der die Umwelt beaufschlagt wird (Schritt 4), um die Zielfunktion zu maximieren. Für die Auswahl gibt es verschiedene Möglichkeiten. Grundlage soll dabei die menschliche Entscheidungsfindung sein, die mit bekannten Strategien der LCS verglichen wird.

Generell werden die Entscheidungsstrategien des Menschen in analytische vs. nichtanalytische sowie in kompensatorische vs. nichtkompensatorische klassifiziert. Bei analytischen werden die Konsequenzen der Optionen als

ausschlaggebendes Entscheidungskriterium miteinander verglichen, während bei nichtanalytischen andere, nichtsachliche Variablen miteinander verglichen werden (z. B. ästhetische). Allerdings müssten nichtanalytische Entscheidungen für den Einsatz in LCS durch die Erstellung einer Zielfunktion zu einer analytischen überführt werden, indem entscheidungsrelevante Variablen in die Zielfunktion integriert werden.

Im vorliegenden Dokument erfolgt die Bewertung der Regeln anhand deren zu erwartenden Gewinn, also $\Psi.r_t$ der Regeln. Denn dieser Wert entspricht dem Zielfunktionswert, in dem beide Teilziele im Smart Home-System mit entsprechender Gewichtung bereits enthalten sind. Damit wird eine analytische Entscheidung getroffen.

Außerdem kann jede Entscheidungsstrategie nach der Fähigkeit zum Kompensieren sortiert werden, wenn zum Beispiel bei einer kompensatorischen Strategie eine gute Konsequenz eine schlechte wettmachen kann. Im vorliegenden Beispiel wäre das der Fall, wenn ein schlechteres Energiemanagement zwar zu einem größeren Energieverbrauch führt, dieser Nachteil aber durch das sehr gut vorhergesagte Verbraucherverhalten wettgemacht würde. Das ist per se in der Definition der Zielfunktion als Summe der beiden (kompensatorischen) Teilziele aus und kann zu paretooptimalen Lösungen führen. Bei nichtkompensatorischen Entscheidungsstrategien müssten Energieminimierung sowie Komfortmaximierung getrennt voneinander verglichen, also zwei Zielfunktionen für jede Regel gebildet und ggf. multiplikativ miteinander verknüpft werden. Der Vorteil dabei wäre, dass eine dezierte Auswertung der Antwort der Umwelt (r_{t1}) in Bezug auf die Entwicklung der Wissensbasis möglich ist, da zwei getrennte Güte- bzw. Zielfunktionen belehrt würden, nachteilig ist die mangelnde absolute Vergleichbarkeit

der Gesamtbelohnungen durch den Anwender. Die Zielfunktion in (63) entspricht einer kompensatorischen Entscheidungsstrategie.

Nachdem klar ist, dass die Entscheidung kompensatorisch und analytisch ist, wird der Bewertungsmaßstab gesucht, um die möglichen Regeln im Action Set miteinander vergleichen zu können. Dazu wird mit der Maximierungsregel die meist bevorzugte Methode herangezogen, nach der die dominante Option, also die Entscheidung, die den größtmöglichen subjektiven Nutzen im Rahmen der vorliegenden Informationen verspricht, gewählt wird. Umso mehr Informationen vorhanden sind, desto mehr Aussagekraft repräsentiert auch die Entscheidung selbst. In der Literatur zur Entscheidungspsychologie führt der Extremfall, dass alle relevanten Informationen vorliegenden, zu einer rationalen Entscheidung, während jeder andere Fall zu einer heuristischen führt. Der subjektive Nutzen ist im vorliegenden Fall die erwartete Belohnung der gewählten Regel $\Psi.r_t$.

Bei klassischen LCS hingegen wird zwar auch mit einer bestimmten Wahrscheinlichkeit η_{\max} (typischerweise über 90 %) die Maximierungsregel einer analytischen, kompensatorischen Entscheidung angewandt, mit der Restwahrscheinlichkeit $1-\eta_{\max}$ jedoch eine beliebige Regel zufällig aus dem vorher gebildeten Match Set gewählt. Da nicht garantiert ist, dass die Regelbasis vollständig den Zusammenhang zwischen Situation s_{t0} und Aktion a_{t1} technisch abbilden kann, sondern nur mit großem Aufwand erfassbare Einflüsse mit kleiner Korrelation eine Rolle spielen können (Rauschen), ist das Modell der handelnden Personen nicht zu 100 % perfekt. Diese Restunsicherheit wird durch die Exploration an dieser Stelle nachgebildet, in deren Folge zufällig eine Regel mit der Wahrscheinlichkeit $1-\eta_{\max}$ gewählt wird.

Bei der menschlichen Entscheidung spiegelt sich diese Tatsache in dem Fakt wider, dass es Menschen ebenfalls unmöglich ist, vor einer Entscheidung

alle möglichen Optionen sowie deren Konsequenzen gedanklich durchzuspielen. Im übertragenen Fall können diese Konsequenzen sogar nur mit einer gewissen Wahrscheinlichkeit ausgemacht werden.

Ins Action Set werden als nächstes alle Regeln übernommen, die die gewählte Aktion a_{t1} ebenfalls vorgeschlagen hätten, da auch deren Selektion zur selben Aktion geführt hätte. Denn die später geerntete Reaktion der Umwelt auf all diese Regeln Einfluss haben. Implementiert ist das wiederum durch ein Bit in jeder Regel („isActionSet“), das zu Beginn jedes Zyklus mit „False“ initialisiert und bei der Übernahme in [A] gesetzt wird.

Beaufschlagung Umwelt und Rückantwort

Die ausgewählte Aktion a_{t1} wird im nächsten Schritt (Schritt 4) ausgeführt. Im Gegensatz zur psychologischen Entscheidung ist in der umgesetzten Implementierung eine Ausführung sicher, nicht lediglich mit einer bestimmten Wahrscheinlichkeit. In der Grundphilosophie des LCS hingegen muss die Umsetzung der Entscheidung nicht vollzogen werden, sondern kann auch in einer „Action List“ gespeichert werden. Das passiert, indem die steuerbaren Geräte im Haushalt entsprechend der Anweisung in a_{t1} real geschaltet werden. Psychologisch ist jetzt wichtig, die menschliche Antwort als Reaktion der Umwelt zu erhalten und auszuwerten.

Nachdem die Nutzerzufriedenheit anhand der korrekt geschalteten Geräte und der Korrekturzeit sowie der Energiebedarf via Messung durch die elektrische Leistungsaufnahme ermittelt wurden, wird die Zielfunktion anhand (63) berechnet und an ENKOS als r_{t1} übermittelt (Schritt 5). Das ist der wesentliche Schritt, um sowohl die Energieminimierung als auch die Komfortmaximierung zu erreichen. Denn der Zielfunktionswert in (63) ist umso

größer, desto besser diese Ziele realisiert sind. Damit erfolgt die Korrektur aller erwarteten Belohnungen, die sich im Updated Action Set $[A]'$ befinden. Die Berechnung ist vom Q-Learning abgeleitet und wird nach (67) durchgeführt. Der Einfachheit halber wurde bei a' und s' die Regelbezeichnung weggelassen, die Werte werden allerdings aus der aktuellen Regelbasis $[P]$ entnommen. Die Formel stellt damit die langfristige Optimierung des Zielfunktionswerts dar.

$$\Psi.r_{t1} = \Psi.r_{t1}(\text{alt}) + \beta \left[r_{t0} + \gamma \bullet \max_a \left(\frac{\sum_{[M]_{a'}} \Psi.r_{t2}(s', a')}{\|[M]_{a'}\|} \right) \right] \quad (67)$$

Danach werden die aktualisierten Regeln (im Schritt 6) zurück in die Population kopiert. In der Implementierung werden damit sowohl $[M]$ als auch $[A]$ und $[A]'$ gelöscht. Dann schließen sich die Generalisierung sowie die Anwendung der Genetischen Algorithmen auf die Wissensbasis an.

6.3.2.2 Die Generalisierung zur Wissensverdichtung (Exploitation)

Die Generalisierung ist das Ersetzen von Werten durch Joker in den Situationsteilen der Regeln. Das bedeutet, dass eine Regel mit zunehmender Generalisierung zu mehr Eingangssituationen passt. Im vorliegenden Fall kann beispielsweise die Entscheidung, das Licht einzuschalten, von der aktuellen Zimmertemperatur abhängen, muss aber nicht. Die Identifizierung der Korrelationsstärke zwischen einem Eingangsparameter und der Zielfunktion, das in der Regelungstechnik als Parameteridentifikation und im maschinellen Lernen als Feature Selection bezeichnet wird, wird auf dem Weg zu-

nehmender Generalisierung sukzessive ermittelt. Wenn nämlich die Regel, die die derzeitige Zimmertemperatur in die Entscheidung für das Lichtschalten einfließen lässt, langfristig erfolgreicher ist als die andere mit einem Joker an der entsprechenden Stelle, dann ist ein Zusammenhang anzunehmen. Das Ziel ist, durch eine maximale Generalisierung die Kausalitäten des Anwenderdenkens so allgemein wie möglich zu identifizieren, aber dennoch die Umwelt bzw. das Verbraucherverhalten so gut wie nötig zu modellieren. Dennoch werden sich erfahrungsgemäß einige spezielle Regeln für Nischenlösungen herausbilden.

Im vorliegenden System wird aus diesem Grund mit einer Wahrscheinlichkeit μ_{Gen} im Bereich $\mu_{\text{Gen}} = 1 \dots 5 \%$, abhängig von Spezifität der Regel, ein Nichtjoker aus der Eingangssituation zu einem Joker umgewandelt. Der Einfluss der Generalisierungswahrscheinlichkeit auf den Erfolg von ENKOS ist später gezeigt, wobei zudem ein Gedächtnis über bereits gegangene, allerdings erfolglose Pfade der Generalisierung aufgebaut wird. Denn sollte ein wesentlicher Zusammenhang durch Fehlgeneralisierung zerstört werden und die damit gebildete Regel nicht erfolgreich sein, wird sich ein anderer, vielleicht Erfolg versprechender Pfad ausbilden.

Da nicht alle Zeichen in der Situation binär (mit Jokern ternär) sind, ist für die Bereiche der Lichtstärke sowie der Uhrzeit die Generalisierung durch die Aufweitung des jeweiligen Bereichs mit einer Wahrscheinlichkeit von $\mu_{\text{Gen,Licht}}$ realisiert. Das bedeutet, dass eine Regel genereller als eine andere ist, wenn diese beispielsweise einen größeren Lichtstärkebereich abdeckt, wenn alle anderen Parameter gleich sind. Dieser Zusammenhang ist in Abbildung 58 verdeutlicht, wo der Bereich an beiden Grenzen durch die Generalisierung erweitert wurde und damit (statistisch) zu mehr Eingangsvektoren s_{t0} passt. Sollte sich der erweiterte Bereich neue Regel Ψ_x in der Praxis

nicht bewähren, weil er beispielsweise keine physikalisch sinnvolle Abbildung der Umwelt darstellt, wird die Regel zwangsläufig wieder aus der Wissensbasis verschwinden. Die Generalisierungsrate des Uhrzeitbereichs wird mit $\mu_{\text{Gen,Uhrzeit}}$ bezeichnet.

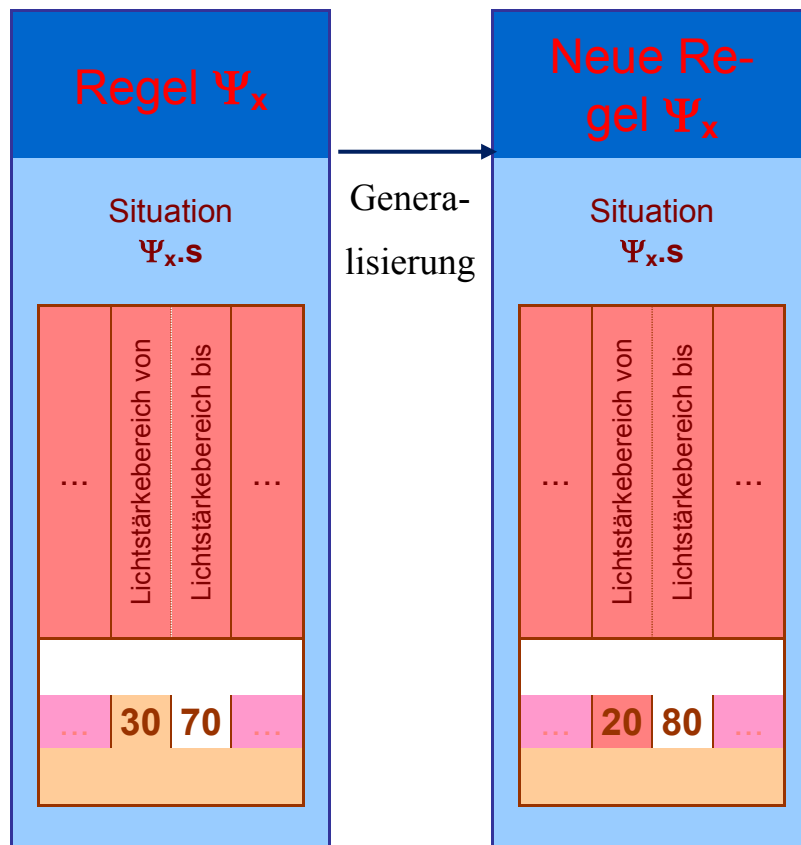


Abbildung 58. Generalisierung des Lichtstärkebereichs einer Regel vom Bereich (30 ... 70) % zum Bereich (20 ... 80) %.

Im Beispiel in Tabelle 12 soll der Zielkonflikt zwischen Generalisierung und zu erwartendem Reward verdeutlicht werden. Während der Reward von Generation 1 über Generation 2 zu Generation 3 sinkt (die Regel wurde zweimal – d. h. über zwei Generationen hinweg – generalisiert), ist die Regel jedoch durch die zunehmende Anzahl der Joker für ein breiteres Eingangsspektrum anwendbar. Das entspricht – bis zu einem gewissen Grad – einer Verbesserung der Regel. Bei menschlichen Verhaltensweisen könnte man

das Beispiel des Smart Home heranziehen. In unterschiedlichen Umweltsituationen sind grundsätzliche Verhaltensmuster erkennbar (beispielsweise das Licht einschalten, wenn es dunkel wird), die auf verschiedene Zimmer oder verschiedene Jahreszeiten angewandt werden können. Diese werden hier als größere Blöcke dargestellt.

Generation	Parameter 1	Parameter 2	Parameter 3	Reward
1	0	1	0	870
2	0	#	0	780
3	#	#	0	500

Tabelle 12. Evolution (Generalisierung) einer Regel, die aus drei Parametern besteht.

Der Unterschied einer generalisierten Regelbasis zu zwei Zeitpunkten ist in Abbildung 59 illustriert. Auf der linken Seite ist eine zerklüftete Regelbasis mit scharfen Trennungen gezeigt, während auf der rechten Seite Regeln nach und nach zu Blöcken zusammengefasst werden, die wiederum einen deutlich größeren Zustandsraum der Umwelt abbilden und an den Grenzflächen ineinander übergehen. So werden die Übergänge einzelner Regeln innerhalb dieser Blöcke fließend, bis irgendwann nur noch die Grundfarben als Blöcke vorhanden sind. Im angegebenen Beispiel würde das einen großen gelben, einen grünen, einen blauen sowie einen lila Block bedeuten, um die gesamte Umwelt beispielsweise mit vier Regeln (statt links 18 sichtbaren) abzubilden, wobei sich kleine Nischen für ganz spezielle, von allen anderen unabhängige Situationen ausbilden werden. Gleichzeitig wird angestrebt, trotzdem einen möglichst großen Reward von der Umwelt zu erhalten.

ten. Das Ziel ist ein stabiler Zustand, in dem die durchschnittliche Spezifität über alle Regeln und der durchschnittliche Reward ausgeglichen sind.

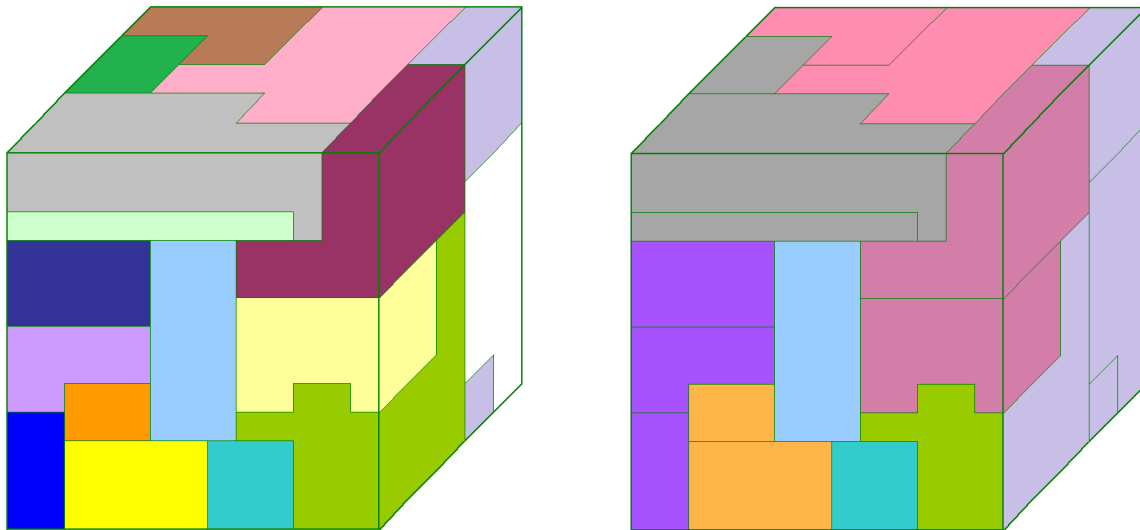


Abbildung 59. Exemplarische Darstellung der Wissensverdichtung. Während auf der linken Seite noch strikt voneinander getrennte Regeln (jeweils eine Farbe) kleine Bereiche des Zustandsraums umfassen, verschwimmen die Regeln auf der rechten Seite zu größeren Blöcken (Blockbildung). Damit verschwimmen auch die Grenzen, womit die Regeln Kompromisse darstellen und größere Bereiche des Zustandsraums abdecken.

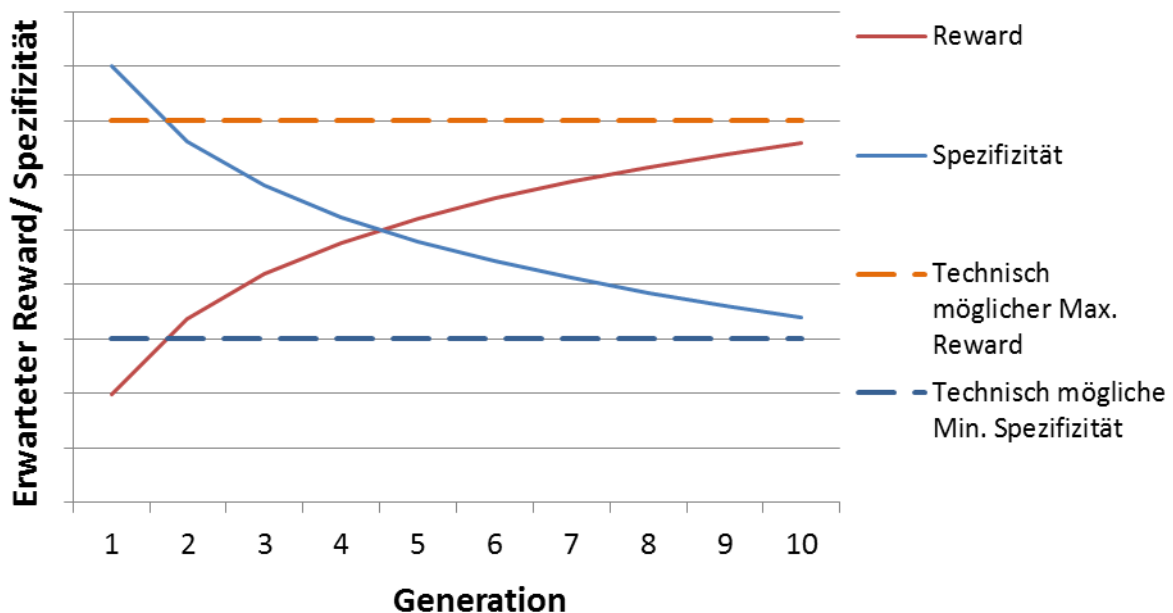


Abbildung 60. Reward und Spezifität über alle Regeln bei jeder Generation.

Wann immer sich der Charakter der Umwelt signifikant ändert, wird sich die Regelbasis anpassen, wie in Abbildung 60 gezeigt ist, bis ein stabiler Zustand erreicht ist. Dieser kann durch die minimal technisch mögliche Spezifität und den maximal technisch möglichen Reward beschrieben werden, vergleichbar mit dem technisch maximal erreichbaren Wirkungsgrad eines Systems. In der Regel bedarf es dafür viele Generationen (s. Abbildung 61).

Situation s					
u(t)					
# P_Bad	# P_Az	Lichtstärke		Uhrzeit	
		von	bis	von	bis
0	0	7	1	2	3
0	#	7	1	#	3
0	#	7	1	#	3
0	#	7	1	#	#
0	#	7	1	#	#
0	#	7	1	#	#
0	#	7	1	#	#
0	#	7	1	#	#
0	#	7	1	#	#
0	#	0	1	#	#

Abbildung 61. Die Generalisierung einer Regel. Im Beispiel ist einerseits die Lichtstärke schrittweise generalisiert worden, während der Zeitstempel zuerst generalisiert und zwischendurch einmal mutiert. Ein Zeitschritt entspricht dabei einem Halbstundenintervall [27].

Die Spezialisierung – als Gegenstück zur Generalisierung – ist prinzipiell durch das Covering möglich. Denn wenn sich die Evolution der Regelbasis infolge von zu viel Generalisierung in eine Sackgasse manövriert haben sollte, sinkt der durchschnittliche Reward des Modells an sich und Regeln müssen durch Covering geschaffen werden, das genau die vorgegebene Situation repräsentiert. In diesem Fall ist ein Lernen durch die Vorgabe des Nutzers

vorgesehen, wenngleich eine Menge an Erfahrung in den Regeln steckt. Deshalb ist prinzipiell als Alternative eine Degeneralisierung denkbar, mit deren Hilfe man die Pfade in der evolutionären Historie einer Regelbasis zurückläuft. Da kein sinnvoller Anwendungsfall bei den menschlichen Verhaltensweisen identifiziert wurde, wurde dieser Ansatz hier nicht umgesetzt.

6.3.2.3 Umsetzung der Evolutionären Algorithmen zur Exploration

Die Evolutionären Algorithmen sind vergleichbar mit der Verallgemeinerung bisheriger und der Erschließung neuer Wissensbereiche durch Kombination vorhandener Erfahrungen sowie der Ableitung kausaler Zusammenhänge im menschlichen Denken. Denn im Rahmen des Lernens bauen sich Menschen ebenfalls sukzessive ein Modell von deren Umwelt auf, aus dem jederzeit Regeln für aktuelle Entscheidungen abgerufen werden.

Die Selektion stellt dabei die Reaktion auf eine sich verändernde Umwelt sowie der Steigerung des Reifegrads (durch Selektionsdruck) dar. Regeln, die nicht länger mithalten können, um die Umwelt mit großen zu erwartenden Reward zu modellieren, werden aussterben und neue, besser angepasste an deren Stelle treten. Damit wird unser Modell von unserer Umgebung in verschiedenen Ebenen der Wissensbasis permanent überarbeitet.

Die Mutation der Regeln spiegelt im Wesentlichen menschliche Neugier auf der einen und unsere (intuitive) Modellverifikation auf der anderen Seite wider. Bei Kindern ist das Modell der Umwelt noch nicht so komplex angelegt wie bei Erwachsenen, das Austesten unbekannter Wege ist deshalb essentieller für deren Weiterentwicklung. Implementiert ist die Mutation durch eine steuerbare Variable, die die Wahrscheinlichkeit einer zufälligen Änderung eines Bits im Situationsteil einer Regel ausdrückt.

Die Rekombination ist die Umsortierung vorhandenen, bereits verifizierten Wissens (zur Steigerung des zu erwartenden Rewards), um dadurch Kausalitäten miteinander zu kombinieren, um damit einen besseren Arbeitspunkt zu finden. Dabei werden Bitstrings zwischen verschiedenen Regeln ausgetauscht. Es wird – wie in der Herleitung der Building Block-Hypothese gezeigt – der vorhandene, in der Regel multidimensionale Lösungsraum als sehr große Anzahl sehr kleiner gültiger Lösungen interpretiert, die jeweils unterschiedlich große Schnittmengen miteinander haben. Diejenigen ohne signifikante Überlappungen sind dabei frei kombinierbar und können unter Umständen in einer alternativen Zusammensetzung ein besseres Ergebnis erzielen. Im täglichen Denken bedeutet dies, dass wir die Waschmaschine wahrscheinlich unabhängig vom Radio im Wohnzimmer benutzen, auch wenn beide in einer Regel zusammengefasst sind. Deshalb führt die Rekombination zur Entkopplung des Nutzerverhaltens bezüglich der Waschmaschine und des Radios.

Die Rekombination kann biologisch als Neuordnung zusammengehöriger Genblöcke interpretiert werden (vgl. Abbildung 62). Zusammengehörig sind in der vorliegenden Arbeit beispielsweise Geräte gleicher Serviceklasse. Das bedeutet, dass durch die Rekombination die Möglichkeit gegeben ist, eine Information von einem Gerät auf ein anderes, gleichartiges im Modell zu übertragen.

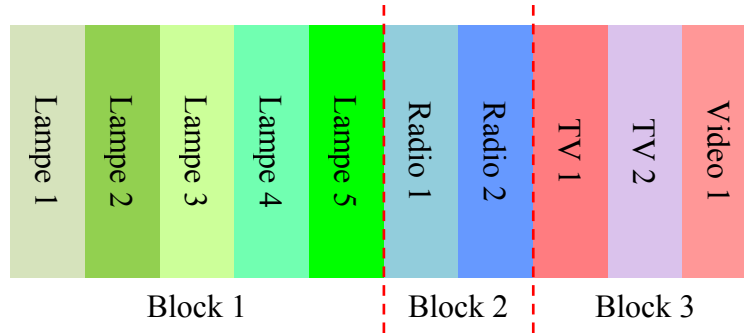


Abbildung 62. Die rot gestrichelten Linien sind die möglichen Stellen für eine Rekombination. Damit ist sichergestellt, dass gleichartige Geräte (i. S. der gleichen Serviceklasse) nicht miteinander kombiniert werden.

Zusammengefasst bilden alle drei Werkzeuge der Genetischen Algorithmen (Mutation, Selektion und Rekombination) wesentliche Vorgänge des menschlichen Denkens ab und kommen im vorliegenden System zur Anwendung. Desweiteren wird deren Einfluss bezüglich des Zielfunktionswerts untersucht.

Bei der Analyse der Forderungen an ein Modell, die in Abschnitt 2.2 aufgezählt sind, ergibt sich ein positives Ergebnis. Mit Hilfe der Learning Classifier Systems konnte – wie schon durch die Neuronale Netze – der systematische Aufbau sowie die Richtigkeit der Umwelt abgebildet werden, auch die relevanten physikalischen Größen (Umweltbedingungen, Verbraucherverhalten, elektrische Geräte) wurden durch die Regeln korrekt abgebildet. Der Ansatz ist deutlich wirtschaftlicher, weil die Regelbasis online belehrt und benutzt werden kann. Sie stellt das Verbraucherverhalten mit einem gewissen Reifegrad dar und verliert die gespeicherten wesentlichen Informationen nicht durch das Entfernen oder Hinzufügen einer Regel. Die vernachlässigbare Simulationszeit ist wirtschaftlich sogar auf einem Mikrocontroller mit einer Leistungsaufnahme im Mikrowattbereich gut darstellbar und die Wissensbasis ist gut vergleichbar mit vorherigen Versionen.

Der wesentliche Vorteil ist die Klarheit, da die Regeln jederzeit ohne weitere Hilfe interpretiert und manuell auf deren Plausibilität hin überprüft werden können. Damit ist die prinzipielle Struktur von ENKOS festgelegt.

Durch die Konfiguration der Simulationsumgebung (vgl. Kapitel 5 und Abbildung 44) werden die Lernrate β , die Vergessensrate γ , die maximale Anzahl der Regeln $\|[\mathbf{WB}]\|$, die Mutationsrate μ_{Mut} , die Generalisierungsrate μ_{Gen} sowie die Lichtgeneralisierungsrate $\mu_{\text{Gen,Licht}}$ neben der Uhrzeitgeneralisierungsrate $\mu_{\text{Gen,Uhrzeit}}$ gesteuert. Außerdem können die Mutation, die Rekombination, das Covering, die Nichtwahrnehmbarkeitserkennung sowie die Gerätesubstitution und die Widerspruchsunterdrückung jeweils ein- bzw. ausgeschaltet werden.

Parameter	Formelzeichen	Wertebereich	Standard	Einheit
Lernrate: Anpassung der Lernrate für den BBA (vgl. (23))	β	0 ... 5	5	%
Vergessensrate: Wahrscheinlichkeit für die Beibehaltung existierender Regeln (Gedächtnis)	γ	0 ... 5	99	%
Maximale Anzahl der Regeln in der Wissensbasis (Population)	i_{max}	0 ... 150	20	Abs. Wert
Mutationsrate: Wahrscheinlichkeit für eine Punktmutation	μ_{Mut}	0 ... 5	1	%
Generalisierungsrate: Wahrscheinlichkeit für die Konversion eines Bits im Situationsteil zu einem Joker	μ_{Gen}	0 ... 5	1	%

Lichtgeneralisierungsrate: Wahrscheinlichkeit für Erweiterung Lichtbereich (um h') ²⁶	$\mu_{\text{Gen,Licht}}$	0 ... 5	10	%
Uhrzeitgeneralisierungsrate: Wahrscheinlichkeit für Erweiterung Uhrzeitbereich (um h') ²⁷	$\mu_{\text{Gen,Uhrzeit}}$	0 ... 5	10	%
Mutation: Einschalten der Mutation (Ausschalten: $\mu_{\text{Mut}} = 0$ %)		0 = Aus 1 = Ein	1	Logischer Wert
Rekombination: Zulassen der Rekombination der Regeln			1	
Covering: Zulassen von Covering, wenn keine passenden Regeln verfügbar sind			1	
Nichtwahrnehmbarkeitserkennung: Zulassen, ob Nichtwahrnehmbarkeitserkennung aktiv ist			1	
Gerätesubstitution: Zulassen der Substitution von Geräten			1	
Widerspruchsunterdrückung: Zulassen der Widerspruchsunterdrückung			1	

Tabelle 13. Parameter für ENKOS, die per Simulationsumgebung gesteuert werden können. Für alle später nicht explizit angegebenen Parameter werden die Standardwerte verwendet.

²⁶ Wenn beispielsweise eine Regel für einen Lichtbereich von 0 bis 20 % Helligkeit angewandt werden kann, wird mit einer gewissen Wahrscheinlichkeit dieser Bereich generalisiert, indem er auf 0 ... 21 % erweitert wird.

²⁷ Generalisierung des Uhrzeitbereichs (vgl. Lichtgeneralisierungsrate).

6.4 Implementierung Energieverbrauchsminimierung

6.4.1 Modellbasierte Nichtanwesenheitserkennung

Der Grundsatz dieser Methode ist, alle nicht wahrnehmbaren Verbraucher abzuschalten, solange kein Nutzer im Raum ist. So ist das Licht von Lampen entbehrlich, wenn eine Person die Lampe gar nicht mehr sehen kann.

Dabei wird auch bereits mit einem unvollständigen Modell Strom gespart, sobald die erste eindeutige Zuordnung eines Geräts zu einem Zimmer erfolgt ist. Die Zuordnung selbst geschieht über den Aufbau eines logischen Modells der Wohnung, das jedem Gerät ein Zimmer zuordnet, in dem es sich zurzeit befindet.

Das Modell wird sukzessive durch die Beobachtung des Verbraucherverhaltens gelernt. Wenn beispielsweise ein Gerät manuell geschaltet wird, während alle Zimmer bis auf eines leer sind, wird eine eindeutige Zuordnung hergestellt, dass das geschaltete Gerät zum nicht leeren Zimmer gehört. Damit entsteht neben der Wissensbasis im LCS ein zweites, logisches Modell der Zuordnung von Geräten zu Räumen im Haushalt, unabhängig vom Verbraucherverhalten der Anwender.

In Abbildung 63 ist die Implementierung in der Simulation gezeigt, wie das Raummodell sukzessive aufgebaut wird. Während auf der linken Seite noch keines der gezeigten Geräte einem Raum zugeordnet ist (keine Schraffur), sind auf der rechten Seite alle Geräte eindeutig platziert. Beispielsweise wird das Gerät **Lampe_2** genau dann bedient, wenn ausschließlich das Arbeitszimmer besetzt ist, sodass dieses Gerät dem Arbeitszimmer zugeordnet wird.

Gerät	Uhrzeitintervall											
	17:30	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	
	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30	23:00	
Lampe_1						x	x	x	x			
Lampe_2									x	x		
PC_1				x	x	x	x	x	x	x	x	
PC_2 (mobil)						x	x	x	x			
Tisch- lampe									x	x		
Radio			x	x	x				x	x		

Abbildung 63. In der Implementierung von ENKOS wird sukzessive das Wohnungsmodell aufgebaut. Während die Lampe_1, die beiden PCs sowie das Radio eingeschaltet werden, während nur im Bad mindestens eine Person anwesend ist (s. gelber Balken in der Zeile „#Person im Bad“), werden die anderen Geräte benutzt, während nur im Arbeitszimmer jemand anwesend ist. Deshalb wird die erste Gerätegruppe dem Bad zugeordnet und blau schraffiert, die zweite Gruppe rot schraffiert dem Arbeitszimmer, sodass die Schraffierung gleichzeitig die Zuordnung ist.

Dieser Algorithmus wurde gewählt, damit Geräte auch dynamisch anderen Räumen zugeordnet werden können, wenn beispielsweise ein mobiles Endgerät in einem anderen Raum angeschlossen wird. Damit adaptiert sich das Modell selbständig. Außerdem ist dadurch keine manuelle Programmierung, sondern ein einfacher, aber robuster Lernvorgang realisiert.

Damit ist ein Teil der Umwelt durch lineare Beziehungen beschreibbar, sodass ein Vergleich mit weniger analytischen, sondern abstrakteren bzw. komplexeren Lernmethoden nicht sinnvoll ist. Denn dann würden lediglich eindeutige Informationen mit zusätzlichen, nicht nötigen Freiheitsgraden versehen.

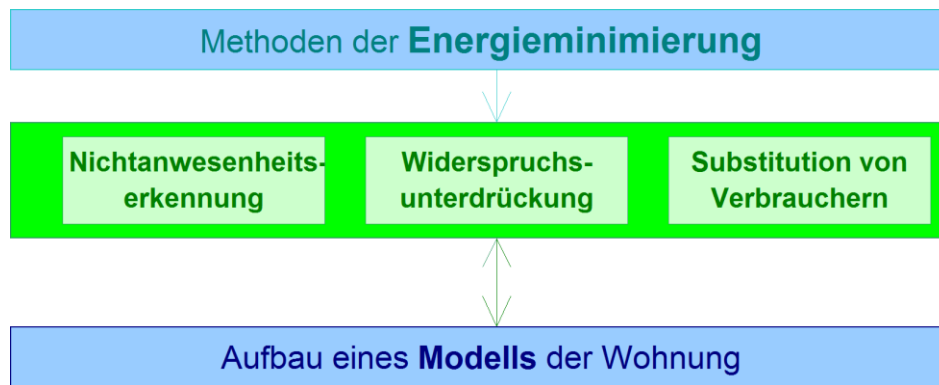


Abbildung 64. Methoden der Energieminimierung sowie Bildung eines physikalischen Modells der Wohnung.

Im nächsten Schritt werden drei Methoden, die die Minimierung des elektrischen Energiebedarfs im Haushalt realisieren, erklärt und in die bereits vorhandene Systemstruktur integriert. Diese Methoden sind die Nichtanwesenheitserkennung, die Widerspruchsunterdrückung sowie die Substitution von Geräten, die in Abbildung 64 dargestellt sind. Diese konkurrieren teilweise mit dem Ziel, den Nutzerkomfort zu maximieren, was eine Gewichtung dieser Teilziele in der Zielfunktion f_{ENKOS} ausbalanciert wird. Das Wechselspiel zwischen analytischen und abstrakten Teilzielen wird später näher beleuchtet.

Als konkrete Implementierung hat sich als beste Lösung die Negativzuordnung gezeigt. Dabei wird zunächst jedes Gerät, bis es eindeutig zugeordnet werden kann, jedem Raum als zugehörig gesetzt. Erst bei einer eindeutigen Abhängigkeit ist eine Zuordnung nur zu einem Zimmer, d. h. die Entkoppelung von allen anderen Räumen, möglich.

Die Integration dieses Ansatzes in die Regelbasis [P] der LCS erfolgt, indem alle Regeln dahingehend modifiziert werden, dass nur Geräte eingeschaltet²⁸

²⁸ Eingeschaltet heißt, dass Geräte dem Nutzer eine Funktion als Service anbieten. Dieses vereinfachte Weltbild unterschlägt, dass es weitere Funktionszustände wie beispielsweise Stand-By gibt. Selbstverständlich können in Abhängigkeit von der Anwesenheit, je nach

sind, die auch wahrgenommen werden können. Diese Prüfung der Regeln ist in Abbildung 65 demonstriert. Hier wurde eine Regel Ψ_x gebildet, die Anwendung findet, wenn unter anderem ein bestimmter Raum leer ist (der aktuelle Zustand des Geräts spielt dabei keine Rolle, da an dieser Stelle ein Joker steht). Die durchzuführende Aktion besagt, dass das Gerät eingeschaltet werden soll.

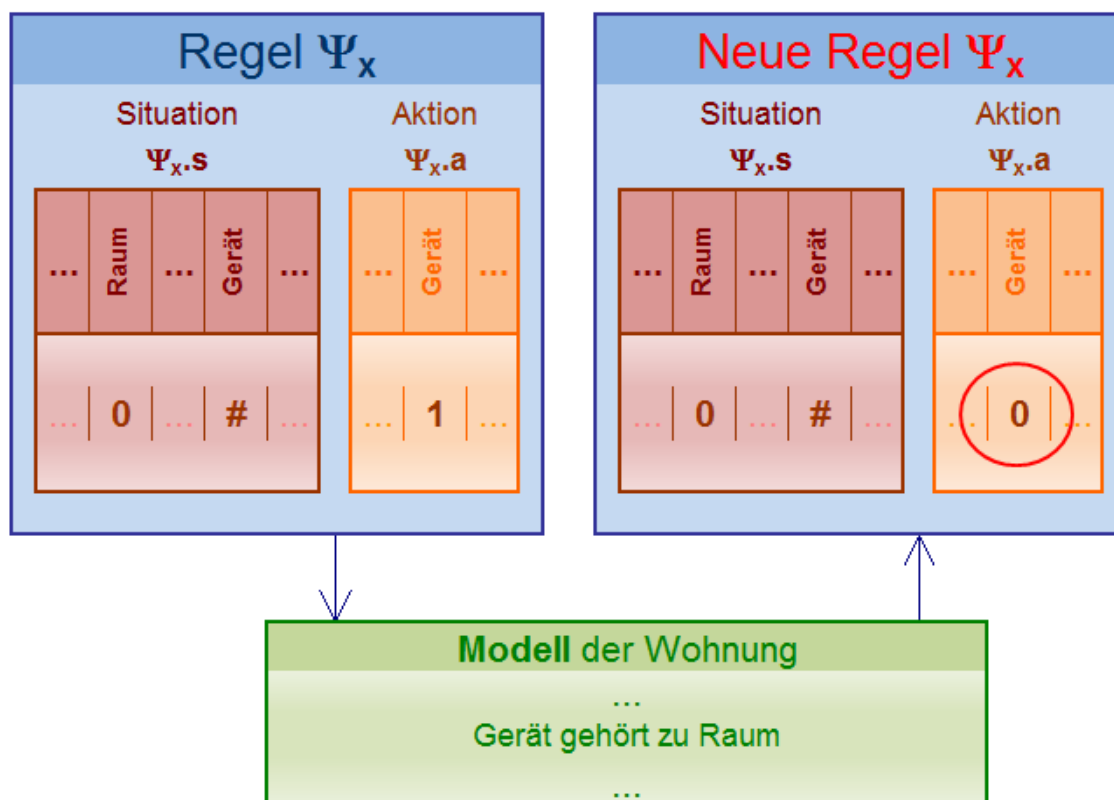


Abbildung 65. Modifikation der Regel Ψ_x infolge der Nichtanwesenheitsprüfung. Das Einschalten von Gerät in der Regel Ψ_x bei einem leeren Raum widerspricht dem Modell der Wohnung. Deshalb wird die Regel Ψ_x mit Hilfe des Wohnungsmodells modifiziert.

Im analytisch gebildeten Modell in Abbildung 65 ist jedoch gelernt worden, dass das einzuschaltende Gerät zum leeren Raum gehört. Deshalb ist ein Einschalten nicht sinnvoll, sodass die Regel entsprechend angepasst wird.

Funktion eines Geräts, auch völlig andere Funktionen sinnvoll sein. Aufgrund der Vielfalt der Möglichkeiten wird darauf verzichtet, da das Grundprinzip das gleiche ist.

Seitenwirkungen werden dabei nicht erwartet, da diese Modifikation unabhängig von anderen Gerätezuständen ist. Die neue Regel Ψ_x ersetzt nun die alte in der Wissensbasis.

Sobald der Raum nicht mehr leer ist, ändert sich der Umweltzustand und eine neue Regel findet Anwendung. Die Integration dieses Modells (implementiert als Tabelle, in der bei jedem Gerät möglichst der Raum, in dem es steht, irgendwann hinterlegt ist) in die Systemtopologie ist in Abbildung 66 gezeigt, wobei das logische Modell mit der Wissensbasis auf der einen Seite und mit dem s_t -Eingangsvektor auf der anderen Seite gekoppelt sind. Damit sind die Belehrung dieses Modells sowie dessen Anwendung möglich.

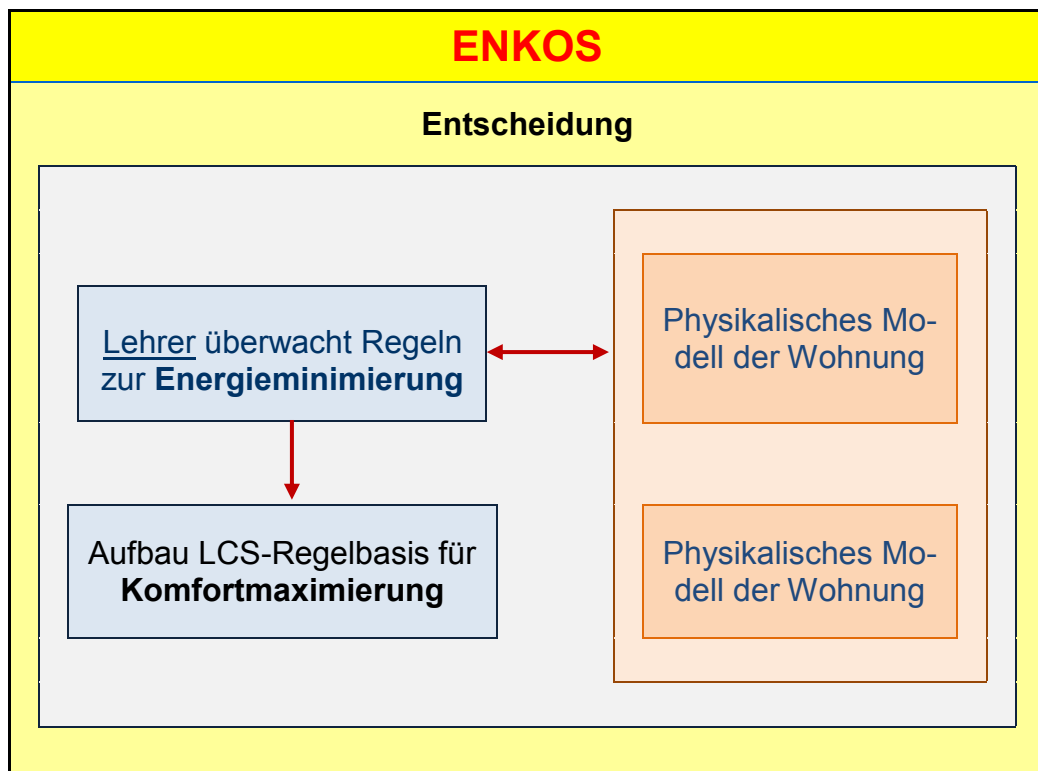


Abbildung 66. Integration des logischen Modells der Raumzuordnung von Geräten in den Block „ENKOS“ (vgl. Abbildung 47). Zur Gewährleistung der Übersichtlichkeit sind die anderen beiden Blöcke des Systems nicht dargestellt.

Um das Energiesparpotential dieser Methode zu ermitteln, wurden jeweils 15 Normaljahre mit und ohne diese Funktion simuliert und in Abbildung 67

dargestellt. Im Durchschnitt ergab sich ein Funktionswert von $f_{\text{ENKOS_ohne}} = 952,07$ (also ohne Integration des Algorithmus‘ zur Anwesenheitserkennung) gegenüber $f_{\text{ENKOS}} = 953,03$.

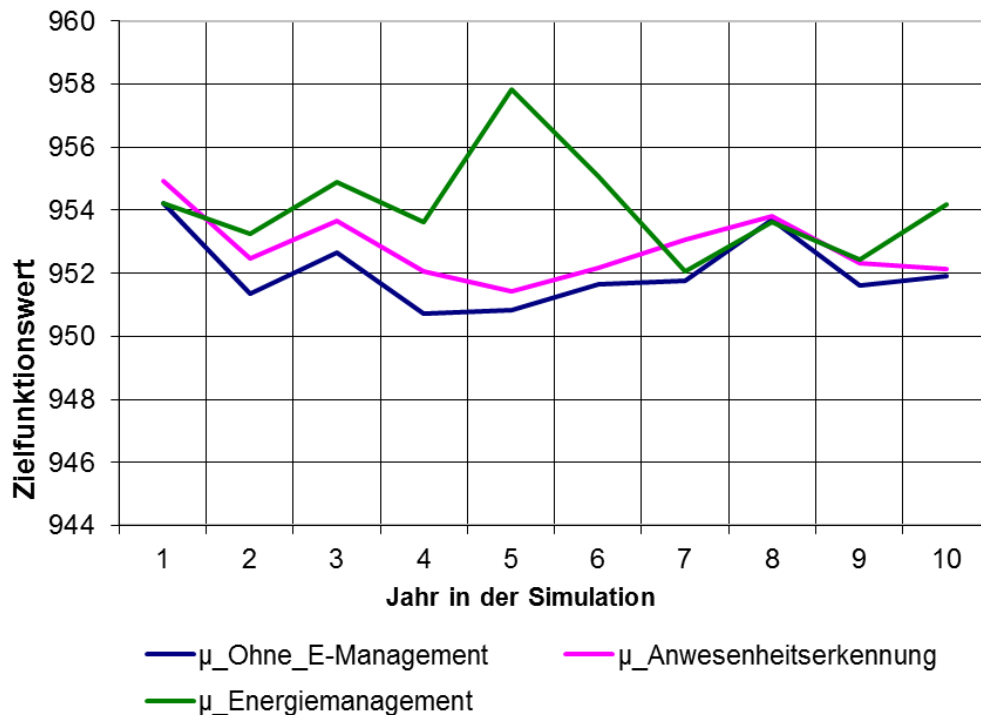


Abbildung 67. Vergleich der Zielfunktionswerte f_{ENKOS} ohne Energiemanagementmaßnahmen im Vergleich mit der aktivierten Anwesenheitserkennung und allen eingeschalteten Energiemanagementmaßnahmen. Es wurden jeweils 10 Jahre viermal nacheinander simuliert und dafür die Mittelwerte benutzt. Die durchschnittliche Standardabweichung aller Messdaten beträgt $s = 3.57$.

Dass der Unterschied nicht deutlicher ausfällt, liegt daran, dass sich in der Zielfunktion auch der Wohnkomfort enthalten ist. Und während der Energieverbrauch sich zwar verringert, sinkt gleichzeitig auch der Wohnkomfort. Eine Gegenüberstellung für den gleichen Zeitraum und die gleichen Simulationseingangsdaten ist in

Abbildung 68 gezeigt. In der Summe jedoch ist eine Verbesserung des Gesamtergebnisses erzielt worden.

Gerät #1	X	X		
Gerät #2		X	X	X
Rewarding Energy consumption $f_{\text{energy}}(t)$	3011	2901	3042	3042
Rewarding prediction $f_{\text{comfort}}(t)$	100	100	92,9	96,4
$f_{\text{ENKOS}}(s,a)$	1,73	1,70	1,52	1,62

Gerät #1	X	X		
Gerät #2		X	X	X
Rewarding Energy consumption $f_{\text{energy}}(t)$	3011	2901	2901	3042
Rewarding prediction $f_{\text{comfort}}(t)$	100	100	100	96,4
$f_{\text{ENKOS}}(s,a)$	1,73	1,70	1,70	1,62

Abbildung 68. Für die gleichen Eingangsdaten ist die Anwesenheitserkennung ein- (linke Seite) und ausgeschaltet. Es ist erkennbar, dass im eingeschalteten Zustand gegenüber dem anderen mehr Energie gespart wird (Rohwert) und damit zu einem größeren Teilfunktionswert $f_{\text{energy}}(t)$ führt, allerdings bei kleinerem Nutzerkomfort.

In der Diskussion des Ergebnisses ist dabei festzustellen, dass die Verhinderung eines allzu großen Verlusts an Nutzerkomfort durchaus einen Mehrenergieverbrauch rechtfertigt. Eine Alternative wäre, die Zielfunktion mit einer weiteren, digitalen Komponente anzureichern, die die Anwesenheitserkennung widerspiegelt. Dieser Weg wird aber nicht realisiert, um eher die realen Nutzerwünsche abzubilden.

6.4.2 Substitution von Verbrauchern

Ansatz dieser Methode der Energieminimierung ist, dass der Mensch nicht primär an der Nutzung eines bestimmten Verbrauchers, sondern an der Bereitstellung dessen Dienstleistung interessiert ist. Das bedeutet, wenn er einen Lichtschalter bedient, möchte er es in erster Linie hell haben. Nachgeordnet wichtig ist dabei, welche der vorhandenen Lampen das Licht bereitstellt.

Bei diesem Ansatz ist eine einfache Lösung implementierbar, wenn zum einen bekannt ist, welchen Service ein bestimmtes Gerät bereitstellt und gleichzeitig, welche Alternativen möglich sind, genau diesen Service zu erhalten. Desweiteren ist für einen Vergleich aller Alternativen miteinander die Information des Energieverbrauchs der Alternativen nötig. Um das zu realisieren, muss das Modell aus Abbildung 65 um die Informationen der Dienstleistungsklasse sowie des Energieverbrauchs eines jeden Geräts erweitert werden, womit die Tabelle 14 entsteht und in ENKOS implementiert wurde. Während die Dienstleistungsklasse selbst eine Information ist, die extern ins Modell integriert werden muss, wird die Energieaufnahme ebenfalls analytisch ermittelt. Dafür sind zwei Möglichkeiten prinzipiell ausprobiert worden:

→ Die eine besteht darin, zwei Umweltzustände s_{t1} und s_{t2} zu finden, die sich lediglich darin unterscheiden, dass das gesuchte Gerät einmal ein- und einmal ausgeschaltet ist (in Abbildung 69 ist als Idealfall diese Differenz genau ein Gerät).

Die Differenz der beiden Leistungsaufnahmen des Gesamtsystems $P(s_{t1})$ und $P(s_{t2})$ lässt sich durch die Integration über eine bestimmte Zeitdauer leicht in den (nominellen) Energiebedarf des gesuchten Geräts $E(\text{Gerät})$ durch (68) ermitteln, indem die elektrische Leistung mit dem normierten Zeitintervall multipliziert wird.

$$P(\text{Gerät}) = P(s_{t1}) - P(s_{t2}) \quad (68)$$

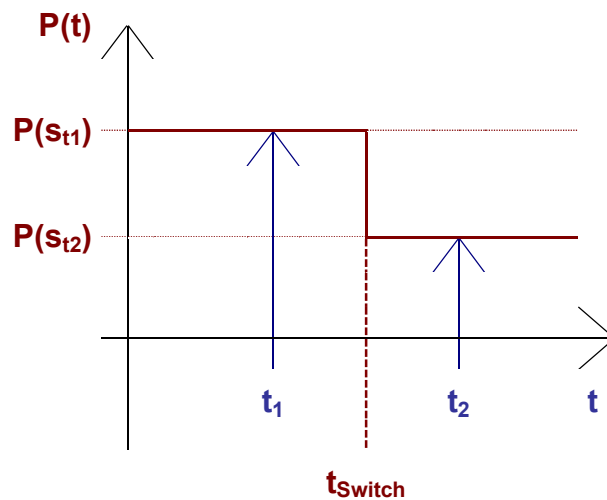


Abbildung 69. Die Methode der zwei Umweltzustände s_{t1} und s_{t2} ist hier dargestellt. Diese Zustände unterscheiden sich darin, dass das Gerät zuerst ein- (s_{t1} zum Zeitpunkt t_1), danach ausgeschaltet ist (Zeitpunkt t_2) mit Differenzleistung $P(t_1) - P(t_2)$.

→ Die zweite Möglichkeit, die Energieverbräuche der Verbraucher zu ermitteln, ist, kurzzeitig durch erzwungene Veränderung des Umweltzustands um genau den gesuchten Verbraucher zu ermitteln. Der Vorteil gegenüber der erstgenannten Methode ist, dass dieser Zustand künstlich generiert werden kann. Der entscheidende Nachteil jedoch ist, dass die Umwelt dadurch in Zustände gebracht wird, die für den Anwender nicht nachvollziehbar und nicht gewünscht sind.

Da sich der Nachteil der zweiten Methode sofort negativ auf die Nutzerzufriedenheit auswirkt, wurde die erste (langwierigere) Methode im finalen System umgesetzt. Während ENKOS diesbezüglich ohne Informationen startet, bilden sich bereits nach wenigen simulierten Wochen die Energieverbräuche heraus.

<i>Anteil am Normmonat</i>	<i>Gerät</i>	<i>Servicegruppe</i>	
		<i>substituierend</i>	<i>behindernd</i>
Arbeitszimmer	Lampe_1	1	1
	Lampe_2	2	2
	PC_1	3	3
	PC_2	4	4
	Tischlampe	5	5
	Radio	6	6
TV/ Audio	TV & Zubehör	7	7
	Badradio	8	8
	Radiowecker1	9	9
	Radiowecker2	10	10
	Audioanlage	11	11
	Küchenradio	12	12
Licht	Flur_1	13	13
	Flur_2	14	14
	Treppe	15	15
	WoZi_1	16	16
	WoZi_2	17	17
	HWR	18	18
	Außen	19	19
	Küche	20	20
	Bad	21	21
	Spie_1	22	22
	Spie_2	23	23
	Dusche	24	24
	Kind_1	25	25
	Kind_2	26	26
	Schlafzimmer	27	27
	Arbeitszimmer	28	28

Tabelle 14. Modell der Dienstleistungsklassen (Initialzustand).

In der Tabelle 14 sind die Dienstleistungsklassen als konkrete Umsetzung gezeigt. Dabei wird jedes Gerät zum einen einer „substituierenden“ Klasse, die in diesem Abschnitt interessant ist, und zum anderen einer „behindernden“ Klasse zugeordnet.

den“ Klasse zugeordnet. Ein elektrischer Verbraucher ist durch einen anderen substituierbar, wenn er dem Nutzer die gleiche Dienstleistung anbieten kann. Im angegebenen Beispiel kann die **Lampe_1** durch die **Tischlampe** ersetzt werden, da beide den gleichen Bereich des Zimmers ausleuchten (können sich gegenseitig substituieren).

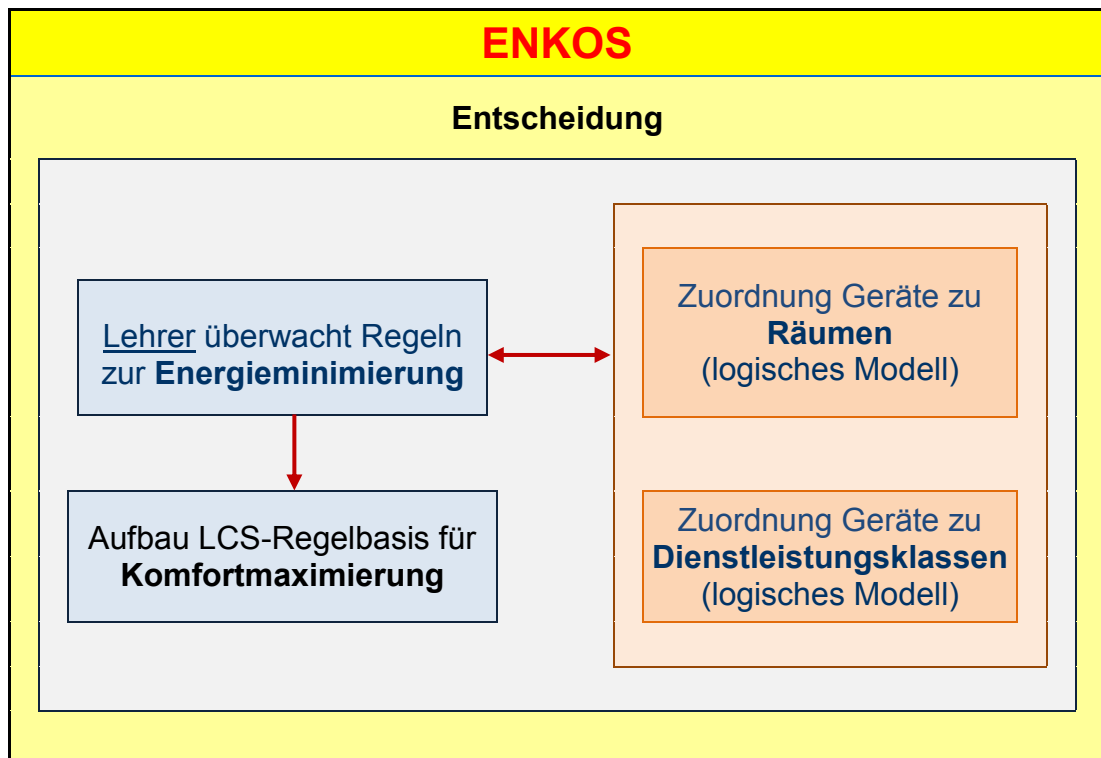


Abbildung 70. Erweiterung des Blocks „ENKOS“ (vgl. Abbildung 47) um die Dienstleistungsklassen, die in Tabelle 14 als Tabelle dargestellt sind, zur Realisierung der Energieminimierung.

Die Integration dieser Tabelle (des Modells) ist in Abbildung 70 gezeigt. Nun sind die Dienstleistungsklassen hinzugekommen, die ebenfalls durch den Eingangsvektor s_t belehrt werden. In Tabelle 14 ist dabei ein Initialzustand gezeigt, der permanent adaptiert wird. Wenn beispielsweise ENKOS eine Lampe unter Benutzung einer sehr spezifischen Regel einschaltet bei der Option einer anderen Lampe, wenn eine generellere Regel benutzt würde, und vom Nutzer dahingehend korrigiert wird, kann davon ausgegangen

werden, dass beide Services („Licht“) zur gleichen Dienstleistungsklasse gehören. Dann konkurrieren zwei Regeln, die einen deutlich anderen Zustandsraum beschreiben, der wiederum den gesuchten Arbeitspunkt jeweils enthält, miteinander. Damit sind zwei Sichtweisen auf die Umwelt miteinander in Einklang gebracht worden, um die Information zu gewinnen, dass Geräte gegeneinander eintauschbar sind.

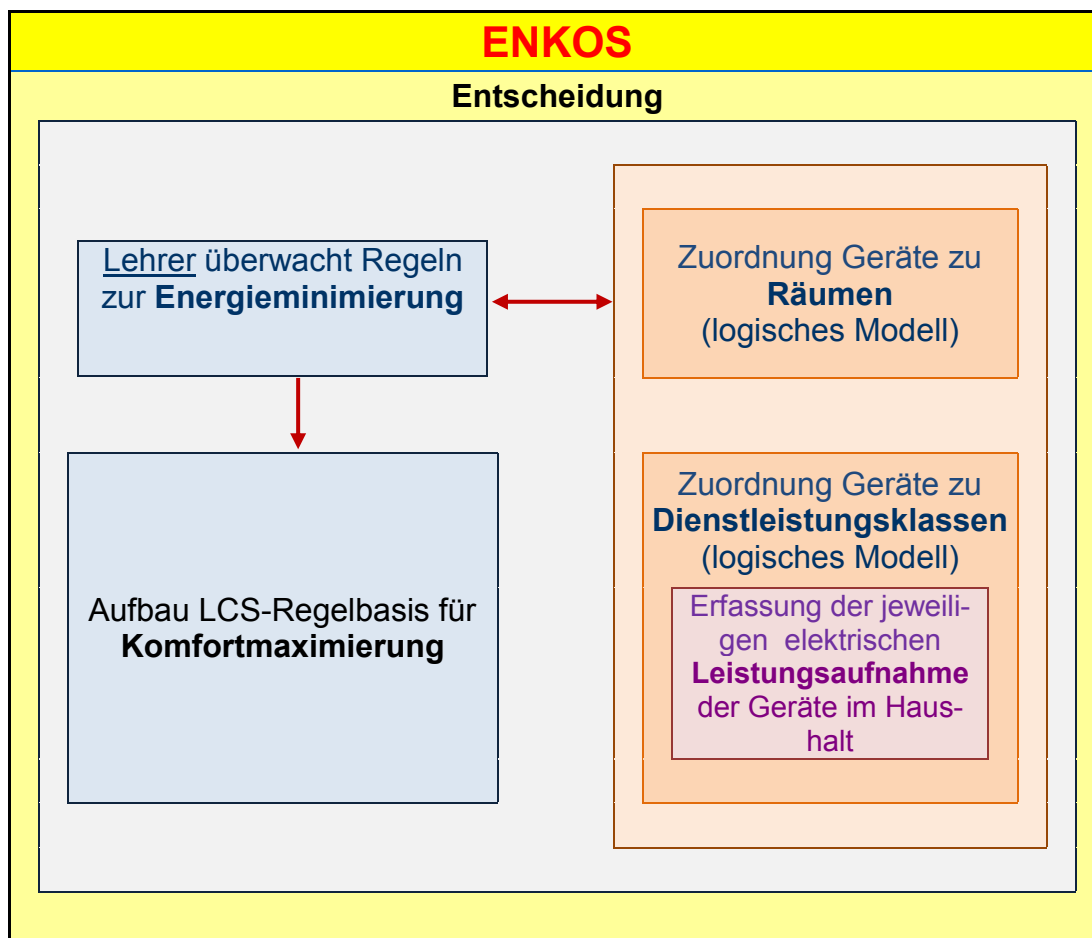


Abbildung 71. Integration der elektrischen Leistungsaufnahme in den Block „ENKOS“ (vgl. Abbildung 47) zur Messung der Gesamtleistungsaufnahme im Haushalt für das elektrische Modell.

Um zu entscheiden, welcher Tausch tatsächlich Sinn ergibt, wird die Information der elektrischen Leistungsaufnahme benötigt, die mit der Methode in Abbildung 69 erhoben wird. Dann konkurrieren „nur“ die geringer werden-

de Nutzerzufriedenheit, wenn ein anderes Gerät eingeschaltet wird als gewünscht mit dem Mehrwert der geringeren elektrischen Leistungsaufnahme. Dieser Konflikt wird über die Gewichtung beider Teilziele in der Zielfunktion gelöst. Die Integration dieser Funktion in die bestehende Systemtopologie ist in Abbildung 71 gezeigt.

Die potentiell eingesparte Energie in der Simulation, wenn diese Technik zum Einsatz kommt, ist in Abbildung 72 gezeigt. Während die blaue Kurve die Simulation eines Referenzhaushalts ohne diese Methode darstellt, zeigt die braune Kurve das Simulationsergebnis der Möglichkeit seitens ENKOS, Services zu substituieren (alle anderen Simulationsparameter sind in Tabelle 13 gezeigt) und die grüne das Ergebnis, wenn alle Energiemanagementmethoden eingeschaltet sind. Dabei sind in fünf unabhängigen Durchläufen jeweils 10 Jahre nacheinander simuliert worden und der Mittelwert wurde gebildet.

Aus den Simulationsergebnissen ergibt sich eine Steigerung von $f_{\text{ENKOS}}(\text{mit_Energiemanagement}) = 953,02$ verglichen mit dem Wert von $f_{\text{ENKOS}}(\text{ohne_Energiemanagement}) = 952,09$ im Durchschnitt von circa 1. Das bedeutet zwar keine überwältigende Verbesserung des Zielfunktionswerts, zeigt allerdings, dass damit eine Verbesserung prinzipiell möglich ist. Desweiteren ist in Abbildung 72 sichtbar, dass die Zielfunktionswerte mit der eingeschalteten Servicesubstitution in bis auf die Jahre 5 und 8 in allen anderen acht simulierten Jahren über denjenigen ohne lagen. Außerdem befanden sich in der simulierten Wohnung keine Geräte von großer Leistungsaufnahme, sondern nur solche im zweistelligen Wattbereich. Wenn das System auf eine Wohnung mit deutlich mehr und energieintensiveren Geräten ausgedehnt wird, sind viel größere Effekte zu erwarten.

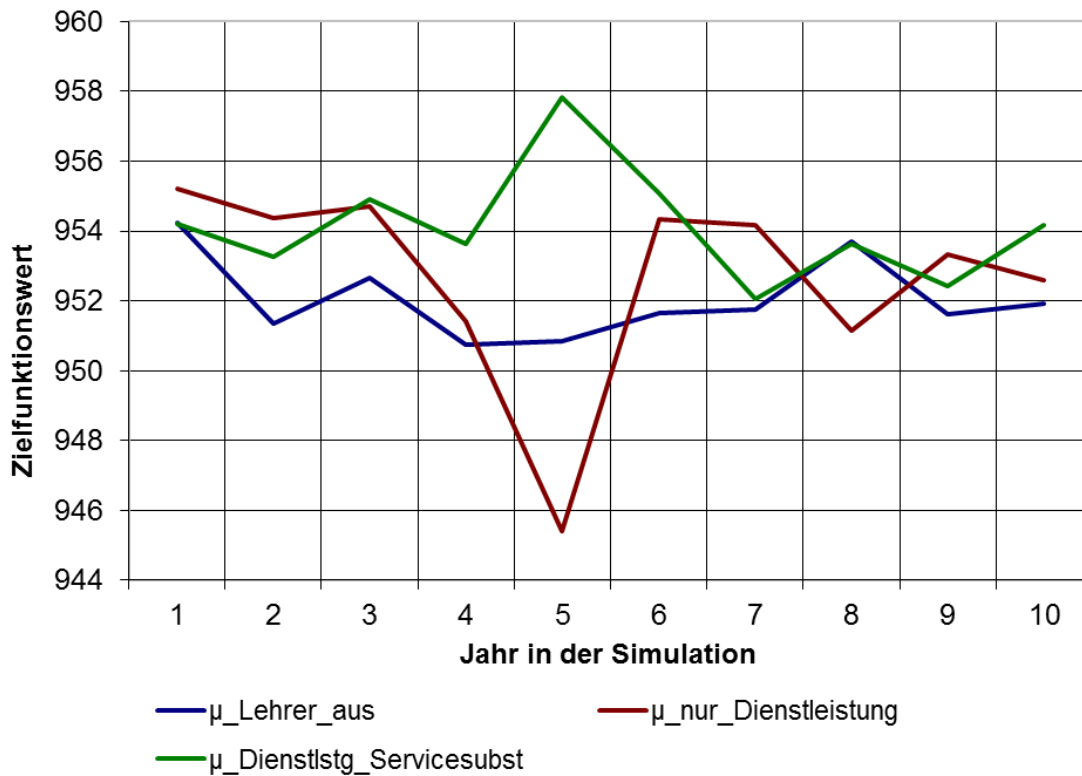


Abbildung 72. Energieeinsparpotential mit der Substitution von Services. Das System (vgl. Abbildung 71) wurde mit denselben Eingangsdaten beaufschlagt, wobei einmal die Lehrerfunktion ein-, einmal ausgeschaltet und einmal nur der Block der „Dienstleistungsklassen“ aktiviert (und die Raumzuordnung abgeschaltet) wurde.

6.4.3 Widerspruchsunterdrückung

Dass der Anwender gleichzeitig nur einen gleichartigen Service (einer „behindernden“ Dienstleistungsklasse²⁹, wie sie in Tabelle 14 dargestellt sind) in einem abgeschlossenen Raum sinnvoll aufnehmen kann, ist der gedankliche Ansatz dieser Methode. Es ist zum Beispiel unsinnig, gleichzeitig Radio und Fernseher mit jeweils eigenem Informationsangebot einzuschalten, die

²⁹ Ein Gerät kann jedoch zu mehreren Dienstleistungsklassen gehören, beispielsweise der Fernseher zu optischen und akustischen Dienstleistern.

sich beide im Wohnzimmer befinden. In einer Konkurrenzsituation ist es nötig, sich dabei für ein Gerät zu entscheiden. Das führt zum Energiesparen um den Betrag der deaktivierten Geräte bei gleichzeitiger Komfortsteigerung. Denn der Anwender spart sich das Abschalten eines eingeschalteten Gerätes, wenn er ein neues Gerät wünscht.

Wenn ein Gerät dem Nutzer mehrere Services anbietet, beispielsweise ein Fernseher, und gleichzeitig mindestens einer, jedoch nicht alle dieser Services einem anderen widersprechen, kann ENKOS nicht eindeutig bestimmen, ob der Widerspruch gelöst werden muss oder nicht. Denn am Beispiel Fernseher kann der Nutzer durchaus das Audiosignal während der Nachrichten nutzen, jedoch auf die optische Anzeige eines PCs (ohne Audioinformationen) nutzen. Dann käme es nicht zu einem Widerspruch per se. Anders herum könnte im gleichen Beispiel eine Energieverschwendung vorhanden sein, wenn der Fernseher voll und ganz in Anspruch genommen würde. Um diesen Konflikt zu lösen, müssen zwingend Annahmen getroffen werden, was in Abbildung 73 dargestellt ist.

Gerät		Konflikt?	
visuell		Nein	
akustisch		Ja	

Abbildung 73. Geräte stehen (teilweise) miteinander in Konflikt, hier am Beispiel des Radios [28] und des Fernsehers [25]. Teilweise bedeutet, dass beide einen akustischen Service anbieten (Konflikt), der Fernseher aber allein ein visuelles Signal (kein Konflikt). Je nach Interpretation können beide aktiviert bleiben, oder auch nicht.

Wenn man dem Prinzip des Energiesparens (Variante 1) den Vorrang einräumt, wird eines der Geräte aus Abbildung 73 abgeschaltet werden. Damit

ist der akustische Konflikt durch Ausschluss gelöst, allerdings unter Verlust der visuellen Informationen. Als Alternative dazu kann die Informationsdichte (Variante 2) im Vordergrund stehen, womit beide Geräte eingeschaltet bleiben würden. Damit ist der Konflikt nicht gelöst, die maximal mögliche Information steht allerdings zur Verfügung. Damit ergeben sich insgesamt drei wesentliche Strategien, um den Konflikt aufzulösen.

Strategie	Vorteil(e)	Nachteil(e)
Das zeitlich zuerst eingeschaltete Gerät wird als erstes abgeschaltet, womit nur der aktuell entstehende Konflikt gelöst wird	Intuitiv dem menschlichen Handeln nachempfunden, da der jüngste Wille als aktueller interpretiert werden kann (vgl. Diskontierung im Q-Learning)	Nur vom menschlichen Denken abgeleitet (keine zwingende Beachtung von Energiesparen und Informationsdichte)
Alle in Konflikt stehenden Services werden durch Abschaltung beseitigt	Maximale Energieeinsparung	Gewünschte Services können verloren gehen (geringe Energiedichte)
Maximalen Nutzerservice bieten, indem keine Konflikte gelöst werden und keine Abschaltung erfolgt	Nutzer erhält immer die Dienstleistung, die er wünscht	Kein Mehrwert durch den Einsatz von ENKOS (→ Medienüberschwemmung)

Tabelle 15. Strategien zum Umgang mit Konflikten bezüglich der Gerätesteuerungen.

Aus den vorgestellten Strategien aus Tabelle 15 wird ersichtlich, dass das Ziel der Energieminimierung hier keinesfalls dem Ziel der Komfortmaxi-

mierung widerspricht, sondern die Vermeidung der Medienüberschwemmung auch den Komfort für den Nutzer steigert.

Um zu entscheiden, welches Gerät im Konfliktfall abgeschaltet wird, wurde sich dafür entschieden, das zuletzt eingeschaltete Gerät als jüngsten Anwenderwunsch zu interpretieren und alle anderen, damit konkurrierenden Geräte zu deaktivieren. Dabei wird zunächst ausgeblendet, dass ein abgeschaltetes Gerät nebenbei einen weiteren, nicht in Konkurrenz stehenden Service, ebenfalls nicht mehr zur Verfügung stellt. Damit sind die zweite und danach das erste Prinzip in Tabelle 15

Als Umsetzung wurden als Methode ebenfalls – wie in Abbildung 65 bereits gezeigt – die Regeln der Wissensbasis entsprechend modifiziert. Damit wird auch diese Methode analytisch integriert. In der Simulation selbst ist das Einsparpotential, das damit erreicht wird, gezeigt. Dabei wurden wiederum fünf Simulationen von jeweils 10 aufeinander folgenden Jahren durchgeführt und dabei sowohl die Widerspruchsunterdrückung aktiviert als auch deaktiviert. Alle anderen Simulationsparameter sind gleich und sind als Standard angenommen.

Die Simulationsergebnisse sind in Abbildung 74 dargestellt. Es stellen wiederum die rote und die grüne Kurve jeweils die deaktivierte und die aktivierte Widerspruchsunterdrückung dar. Es ergibt sich ein Einsparpotential von $\Delta f_{\text{ENKOS}} = 2,03$, wenn die Widerspruchsunterdrückung eingeschaltet wird. Auch hier ist der Effekt nicht allzu groß, trotzdem ist in acht von zehn simulierten Jahren eine Verbesserung des Zielfunktionswerts messbar.

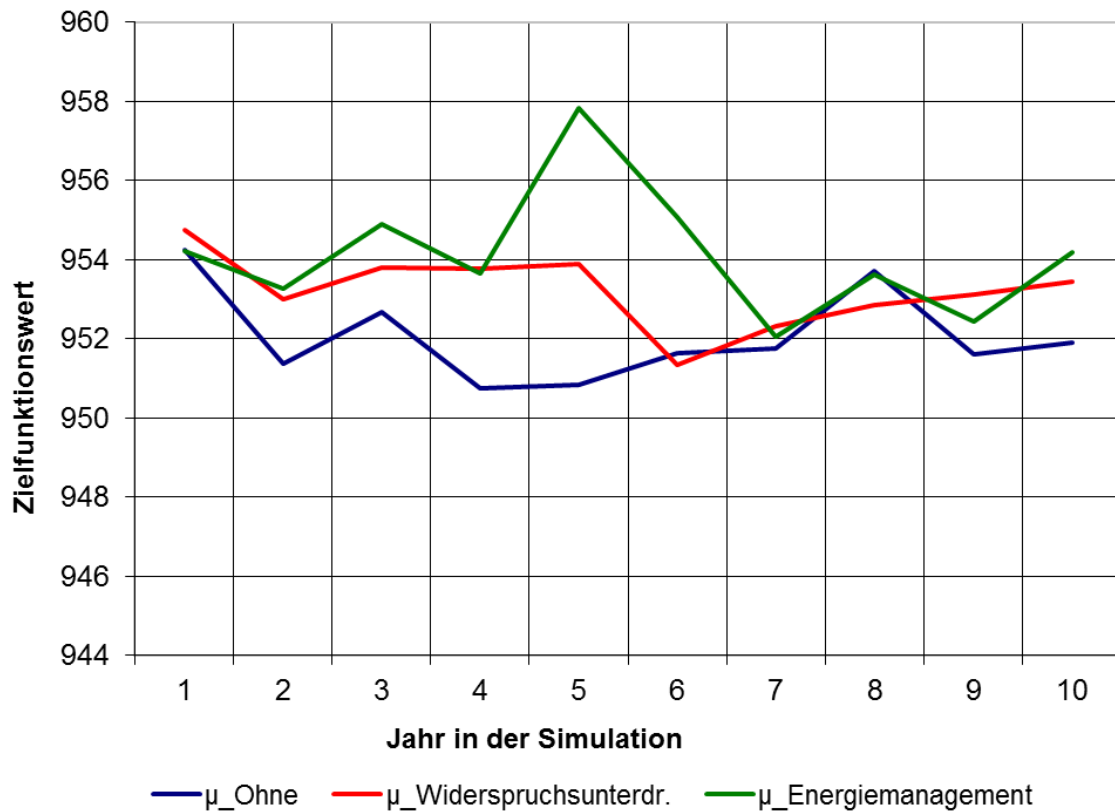


Abbildung 74. Energieeinsparpotential mit der Widerspruchsunterdrückung.

6.4.4 Zusammenfassung der Methoden zur Energieminimierung

Um den Einfluss aller Methoden zur Energieminimierung im Haushalt zu messen, wurden umfangreiche Simulationen durchgeführt, bei denen einzelne Methoden ein- und ausgeschaltet wurden. Als Resultat, das in Tabelle 16 dargestellt ist, hat in der Simulationsumgebung die Nichtanwesenheitserkennung den größten Einfluss auf f_{ENKOS} , während die Servicesubstitution sowie die Widerspruchsunterdrückung für sich genommen nicht allzu viel Verbesserungen bringen. Die Kombination von Nichtanwesenheit und Substitution ist dabei sogar erfolgreicher als alle drei Methoden gemeinsam.

Versuchsreihe	Methoden			f_{Enkos}	Δf_{Enkos}
	Anwesenheitserkennung	Service-substitution	Widerspruchsunterdrückung		
1				952,09 ± 1,11	0,00
2				953,62 ± 1,16	1,53
3				952,67 ± 2,90	0,58
4				952,23 ± 0,94	0,14
5				954,36 ± 1,88	2,27
6				954,12 ± 1,62	2,03

Tabelle 16. Zusammenfassung der Zielfunktionswerte für die einzelnen Methoden zur Energieminimierung. Die grün ausgefüllten Felder markieren den eingeschalteten Zustand der jeweiligen Methode, während in der letzten Spalte die Verbesserung des Zielfunktionswerts infolge der Aktivierung bzw. Deaktivierung einzelner Funktionen quantifiziert wird.

Mit Vorsicht zu genießen sind die absoluten, arithmetischen Mittelwerte der Messreihen, da die Standardabweichung teilweise die errechneten Differenzen zwischen den Versuchsreihen ausfüllt. Trotzdem ist die Tendenz sichtbar, dass ein verstärkter Einsatz der analytischen Methoden den Wert der Zielfunktion steigert.

7 Einfluss verschiedener Modellparameter auf den simulierten Zielfunktionswert

7.1 Einfluss der Lernrate

Während im Kapitel 6 bereits der Einfluss der analytischen Methoden auf den simulierten Zielfunktionswert gezeigt wurde, werden in diesem Abschnitt verschiedene Lernraten auf ihre Verbesserung auf das Ergebnis hin bewertet. In Abbildung 75 sind die Zielfunktionswerte in Abhängigkeit verschiedener Lernraten im Bereich $\beta = [1 \dots 5] \%$ über 10 Jahre simuliert worden. Dabei ist zu erkennen, dass die Streuung etwa gleich groß ist bei jeder Lernrate, sich jedoch qualitativ für kleinere Lernraten größere Zielfunktionswerte ergeben.

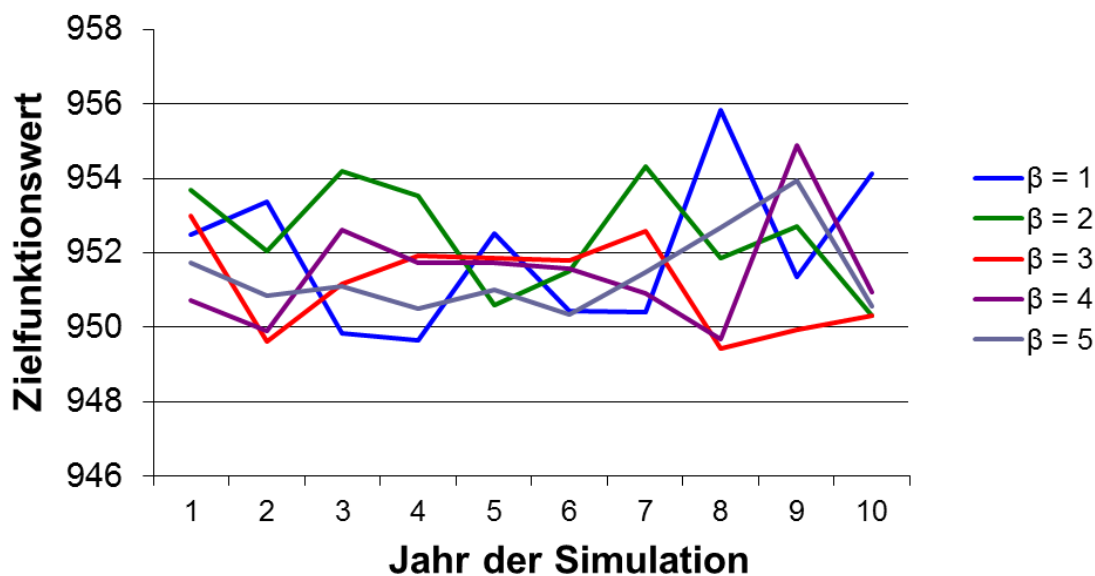


Abbildung 75. Einfluss unterschiedlicher Lernraten β auf den Zielfunktionswert (Simulation von zehn aufeinanderfolgenden Jahren).

Dieser Zusammenhang ist in Tabelle 17 dargestellt. Dabei ist erkennbar, dass das beste Ergebnis bei $\beta = 2 \%$ erzielt wurde. Alle anderen Parameter

wurden dabei konstant gelassen, womit nach der Einfaktormethode gearbeitet wurde [99]. Diese Methode bietet sich in diesem Fall an, da die Anzahl der zu messenden Parameter übersichtlich ist und bislang keine Erfahrungen über ähnliche Systeme bestehen. Deshalb werden auf diesem Weg grundsätzliche Zusammenhänge ermittelt.

Lernrate β in [%]	f_{ENKOS}	
	Mittelwert \pm Standardabweichung	Differenz zum besten Wert
1	952,0 \pm 2,0	0,5
2	952,5 \pm 1,4	0,0
3	951,2 \pm 1,2	1,3
4	951,5 \pm 1,5	1,0
5	951,4 \pm 1,1	1,1

Tabelle 17. Durchschnittswerte der Zielfunktion f_{ENKOS} in Abhängigkeit von der Lernrate β . Für die Lernrate von 2 % ergibt sich der größte Wert der Zielfunktion.

Der Zusammenhang zwischen Lernrate und Zielfunktionswert ist in Abbildung 76 dargestellt. Hierbei ist die Tendenz, dass der Zielfunktionswert bei zunehmender Lernrate (ab $\beta = 2\%$) abnimmt.

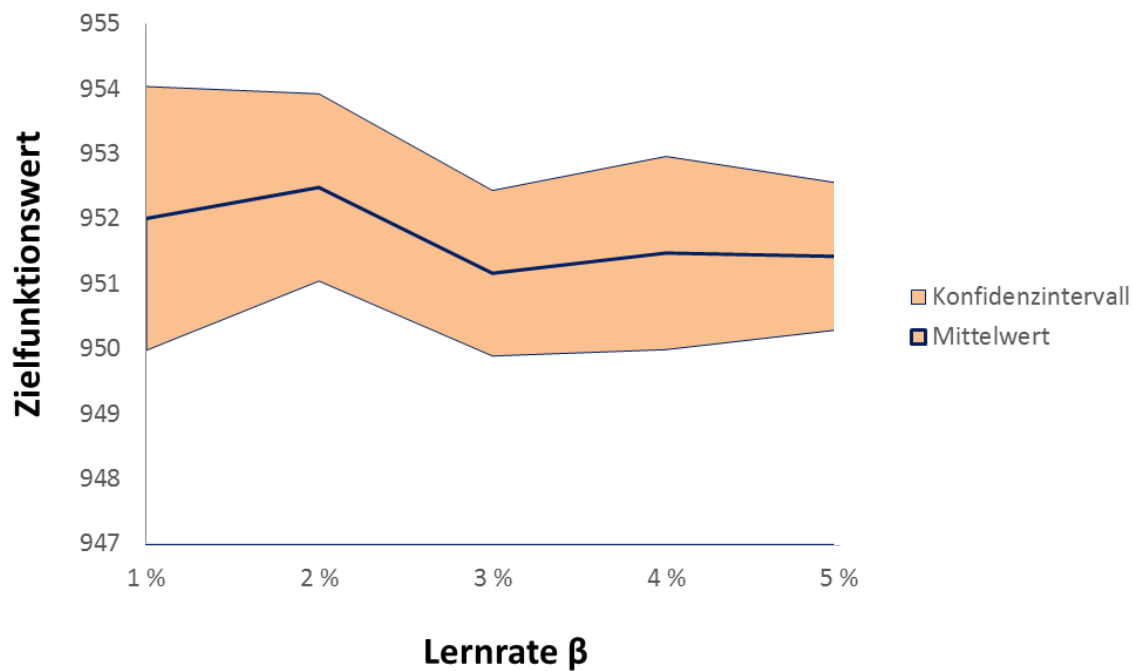


Abbildung 76. Zielfunktionswert in Abhängigkeit unterschiedlicher Lernraten.

Die Hauptwirkung der Lernrate auf den Zielfunktionswert beträgt damit 1,3 (maximale Abweichung) im gegebenen Wertebereich.

7.2 Einfluss der Mutationsrate

Ein wesentlicher Parameter des Genetischen Algorithmus ist die Mutationsrate, mit der zufällig neue Zustände im Situationsraum durchsucht werden. In Abbildung 77 sind verschiedene Zielfunktionswerte für 10 aufeinanderfolgende Jahre simuliert und für verschiedene Mutationsraten dargestellt. Hier ist bereits ersichtlich, dass die Mutationsrate von 1 % die bei jedem einzelnen Jahr die schlechtesten Ergebnisse liefert. Dabei wird scheinbar die Umwelt nur unzureichend nach unbekannten Bereichen durchsucht.

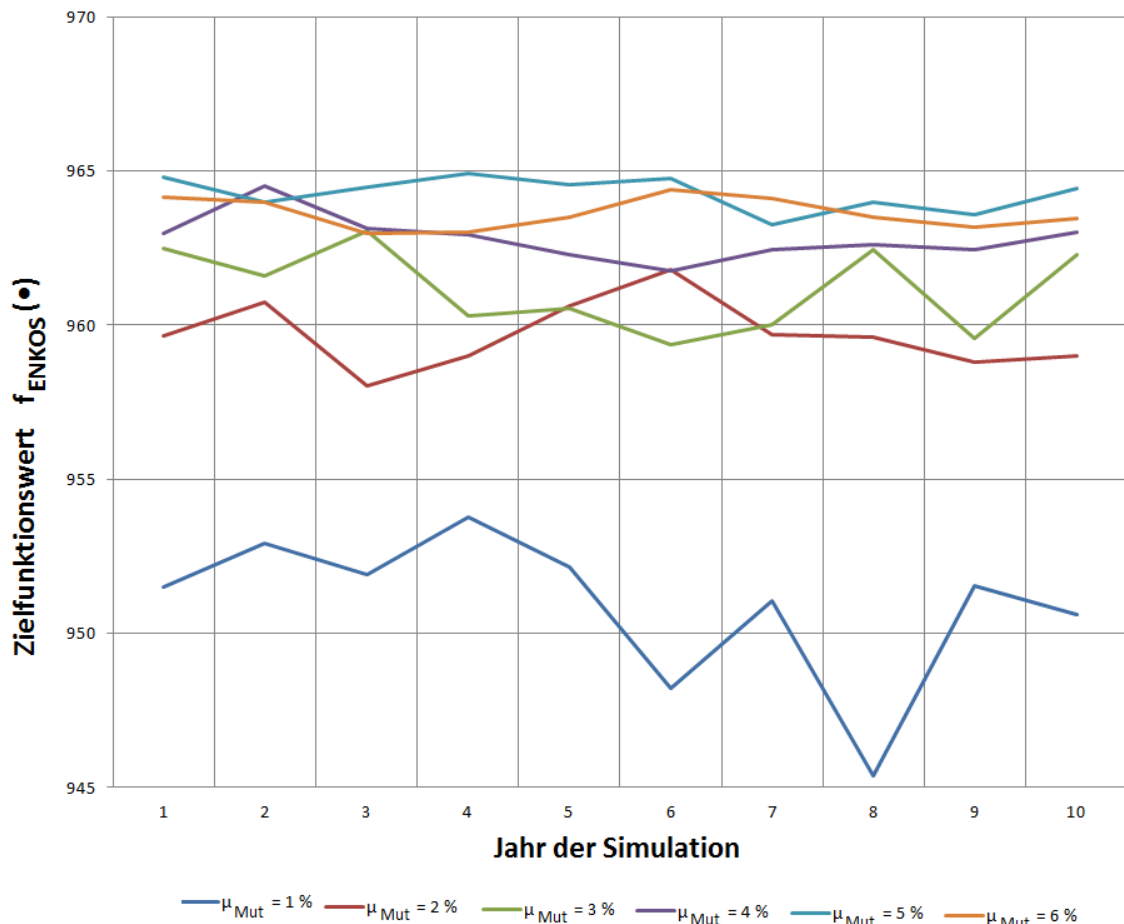


Abbildung 77. Einfluss unterschiedlicher Mutationsraten auf den Zielfunktionswert (Simulationszeit: 10 Jahre).

Die Durchschnittswerte abhängig von der Mutationsrate sind in der Tabelle 18 dargestellt. Während der größte Wert für $\mu_{Mut} = 5\%$ erreicht wird, ist bei einer Mutationsrate von $\mu_{Mut} = 1\%$ der kleinste Zielfunktionswert sichtbar. Bemerkenswert ist ebenfalls die Abnahme der Standardabweichung bei zunehmenden Mutationsraten, was als zunehmende Sättigung des Systems bezüglich dieses Parameters interpretiert werden kann.

Mutationsrate μ_{Mut} in [%]	f_{ENKOS}	
	Mittelwert \pm Standardabweichung	Differenz zum besten Wert
1	950,9 \pm 2,4	13,4
2	959,7 \pm 1,1	4,6
3	961,2 \pm 1,3	3,1
4	962,8 \pm 0,7	1,5
5	964,3 \pm 0,5	0,0
6	963,6 \pm 0,5	0,7

Tabelle 18. Durchschnittswerte der Zielfunktion f_{ENKOS} in Abhängigkeit von der Mutationsrate μ_{Mut} (größter Wert bei 5 %).

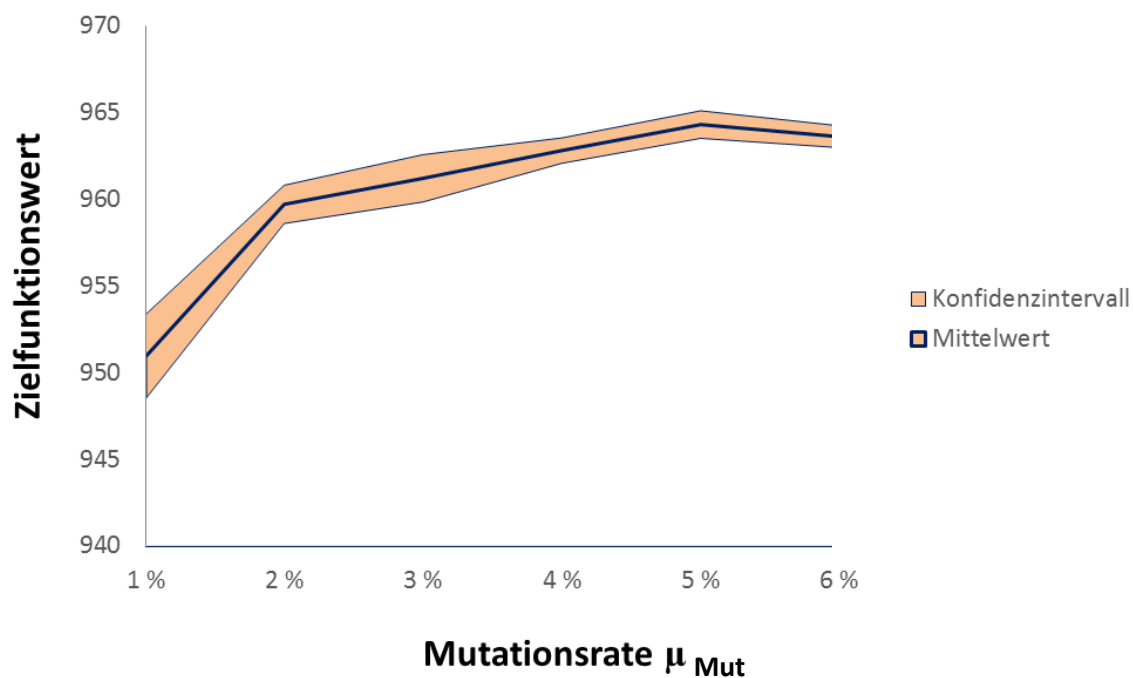


Abbildung 78. Zielfunktionswert in Abhängigkeit unterschiedlicher Mutationsraten (Darstellung Mittelwert sowie Konfidenzintervall nach jeweils 10 Jahren).

Die Abhängigkeit des Zielfunktionswerts von unterschiedlichen Mutationsraten im Bereich von 1 ... 6 % ist in Abbildung 78 visualisiert.

Der Einfluss der Mutationsrate auf das Ergebnis ist, im Gegensatz zur Lernrate, deutlich größer. Zwischen $\mu_{\text{Mut}} = 1 \%$ und $\mu_{\text{Mut}} = 6 \%$ liegt eine Differenz des Mittelwerts von 13,4.

7.3 Einfluss der Generalisierungsrate

Im nächsten Schritt ist der Wert der Zielfunktion in Abhängigkeit von der Generalisierungsrate gezeigt.

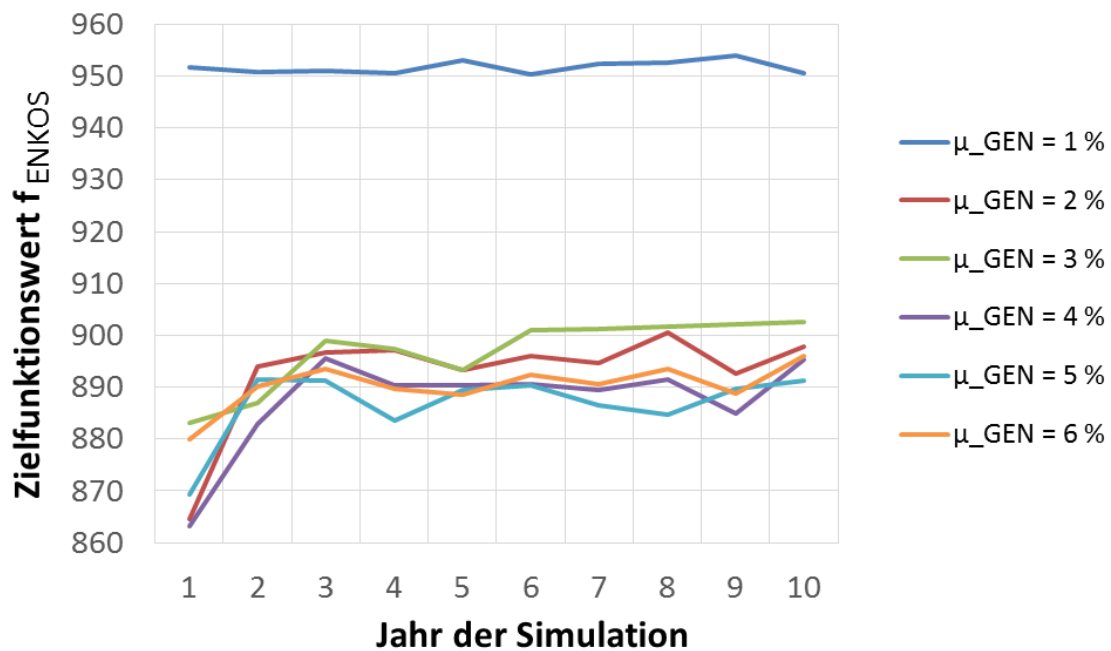


Abbildung 79. Einfluss unterschiedlicher Mutationsraten auf den Zielfunktionswert (Simulationszeit: 10 Jahre).

Hier zeigt sich in Abbildung 79 eindeutig das beste Ergebnis bei einer Generalisierungsrate von 1 %. Das bedeutet, dass die Wissensbasis sich eher allmählich verdichten soll, da sonst eine zu schnelle Blockbildung einsetzt.

Desweiteren wird mit einem Durchschnittswert von $f_{\text{ENKOS}} = 951,7$ ein sehr großer Wert erreicht.

Generalisierungsrate μ_{Gen} in [%]	f_{ENKOS}	
	Mittelwert \pm Standardabweichung	Differenz zum besten Wert
1	951,7 \pm 1,24	0,0
2	892,7 \pm 10,14	59,0
3	896,8 \pm 6,89	54,9
4	887,4 \pm 9,40	64,3
5	886,8 \pm 6,76	64,9
6	890,4 \pm 4,33	61,4

Tabelle 19. Durchschnittswerte der Zielfunktion f_{ENKOS} in Abhängigkeit von der Generalisierungsrate μ_{Gen} (größter Wert bei 1 %).

Während im ersten Jahr die Zielfunktionswerte für alle Generalisierungsraten größer als 1 % noch schlecht sind, werden diese allmählich besser. Da allerdings bei 1 % ein gleichbleibend großer Wert erreicht wird, ist auch keine dynamische Anpassung sinnvoll (vgl. Abbildung 80).

Hier beträgt der Einfluss auf den Zielfunktionswert zwischen dem am schlechtesten und am besten gewählten Parameter 64,9 im Definitionsbereich von 1 bis 6 %.

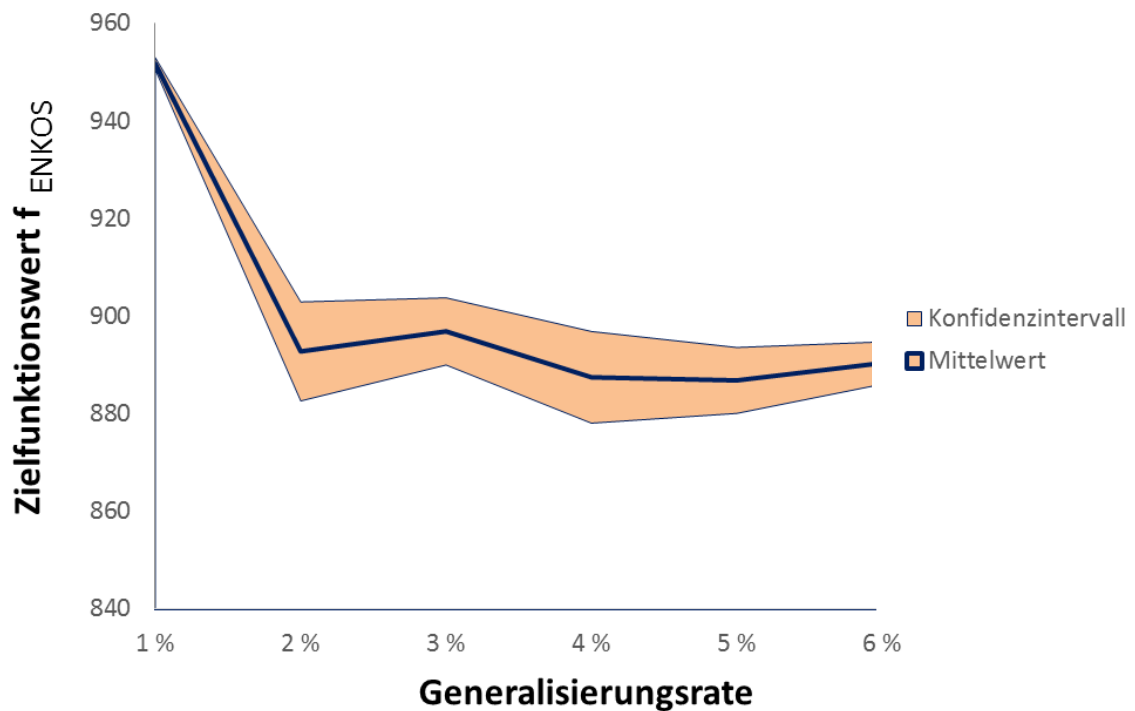


Abbildung 80. Zielfunktionswert bei unterschiedlichen Generalisierungsraten (Darstellung Mittelwert sowie des Konfidenzintervall nach jeweils 5 Messreihen mit 10 Jahren).

7.4 Einfluss der Generalisierungsrate der Lichtstärke

Im Unterschied zur Generalisierungsrate der einzelnen Bits beschreibt die Generalisierungsrate für die Lichtstärke $\mu_{\text{Gen,Licht}}$ die Wahrscheinlichkeit, dass der erfasste Bereich der Lichtstärke erweitert wird. Das bedeutet, dass die Bedingung, wenn alle anderen Parameter gleich bleiben, zu einem größeren Lichtstärkenbereich passt.

Wie in Abbildung 81 und Abbildung 82 ersichtlich ist, hat dieser Parameter kaum Einfluss auf den Zielfunktionswert f_{ENKOS} . Trotzdem ist die leichte Tendenz erkennbar, dass eine Zunahme der Generalisierungsrate zu einer Verschlechterung der Leistungsfähigkeit des Systems führt. Der beste Wert ergibt sich für die kleinste Generalisierungsrate von $\mu_{\text{Gen,Licht}} = 0 \%$, was der

Deaktivierung dieser Funktion entspricht. Das bedeutet, dass die Generalisierung dieses Teils der Regeln nicht sinnvoll ist.

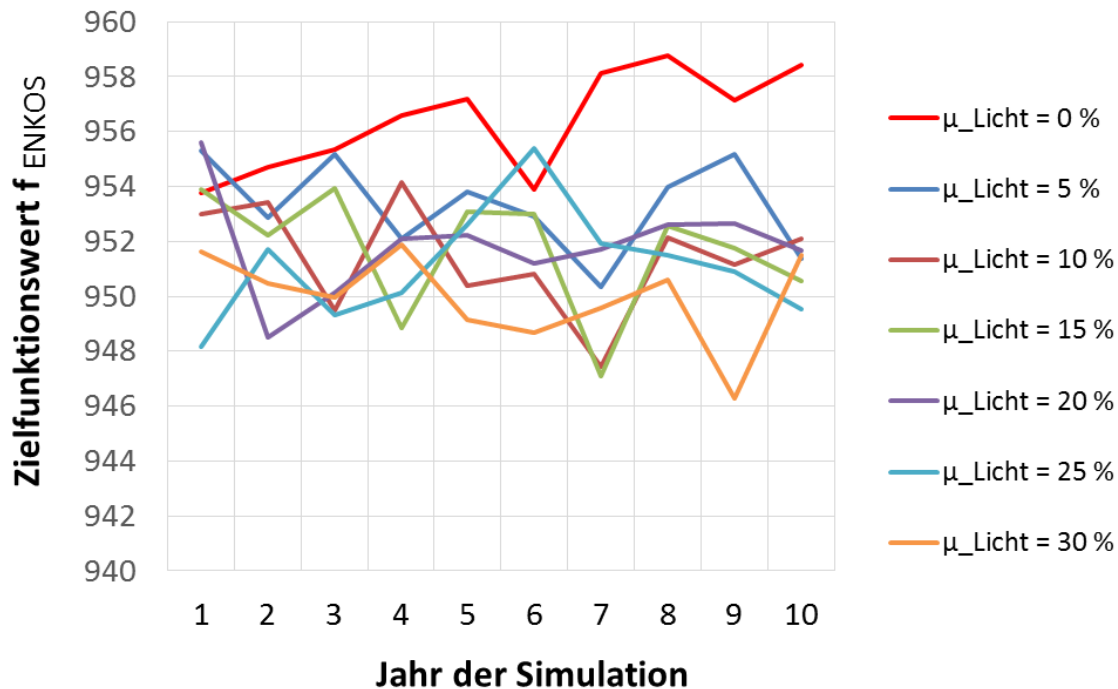


Abbildung 81. Zielfunktionswert in Abhängigkeit unterschiedlicher Generalisierungsraten der Lichtstärke $\mu_{\text{Gen,Licht}}$ über 10 simulierte Jahre.

Im Gegensatz zur allgemeinen Generalisierungsrate werden die Intervalle zum einen in 5 %-Schritten gewählt und beginnen zum anderen bei 0 %. Hintergrund ist, dass diese Art der Generalisierung von reellwertigen Bereichen in dieser Art in der Literatur nicht gefunden wurde. Deshalb soll ein größerer Bereich durchsucht werden, aber auch eine völlige Deaktivierung (= 0 %) möglich sein.

Generalisierungsrate $\mu_{\text{Gen,Licht}}$ in [%]	f_{ENKOS}	
	Mittelwert \pm Standardabweichung	Differenz zum besten Wert
0	956,4 \pm 1,86	0,0
5	953,3 \pm 1,70	3,1
10	951,4 \pm 2,00	5,0
15	951,7 \pm 2,23	4,7
20	951,8 \pm 1,84	4,5
25	951,1 \pm 2,03	5,3
30	950,0 \pm 1,67	6,4

Tabelle 20. Durchschnittswerte der Zielfunktion f_{ENKOS} in Abhängigkeit von der Generalisierungsrate der Lichtstärke $\mu_{\text{Gen,Licht}}$. Für die grün hinterlegte Generalisierungsrate von 0 % ergibt sich der größte Wert der Zielfunktion.

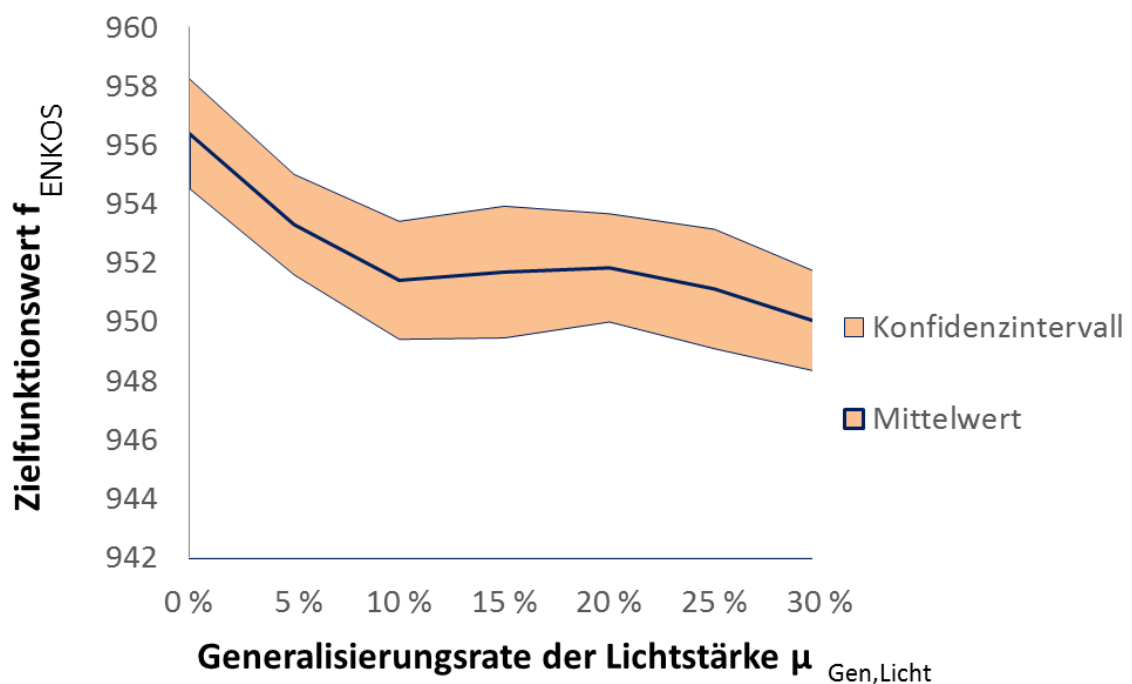


Abbildung 82. Zielfunktionswert in Abhängigkeit unterschiedlicher Generalisierungsraten der Lichtstärke $\mu_{\text{Gen,Licht}}$ (Darstellung des Mittelwerts sowie des Konfidenzintervalls nach jeweils 5 Messreihen mit 10 Jahren).

Tatsächlich zeigt sich das beste Ergebnis, wenn die Möglichkeit, den Lichtstärkebereich für Regeln zu erweitern, ausgeschaltet wird, wie in Tabelle 20

7.5 Einfluss der Generalisierung des Uhrzeitbereichs

Im nächsten Schritt wird Zielfunktionswert in Bezug auf den Uhrzeitbereich, der ebenfalls reellwertig implementiert wird, ermittelt.

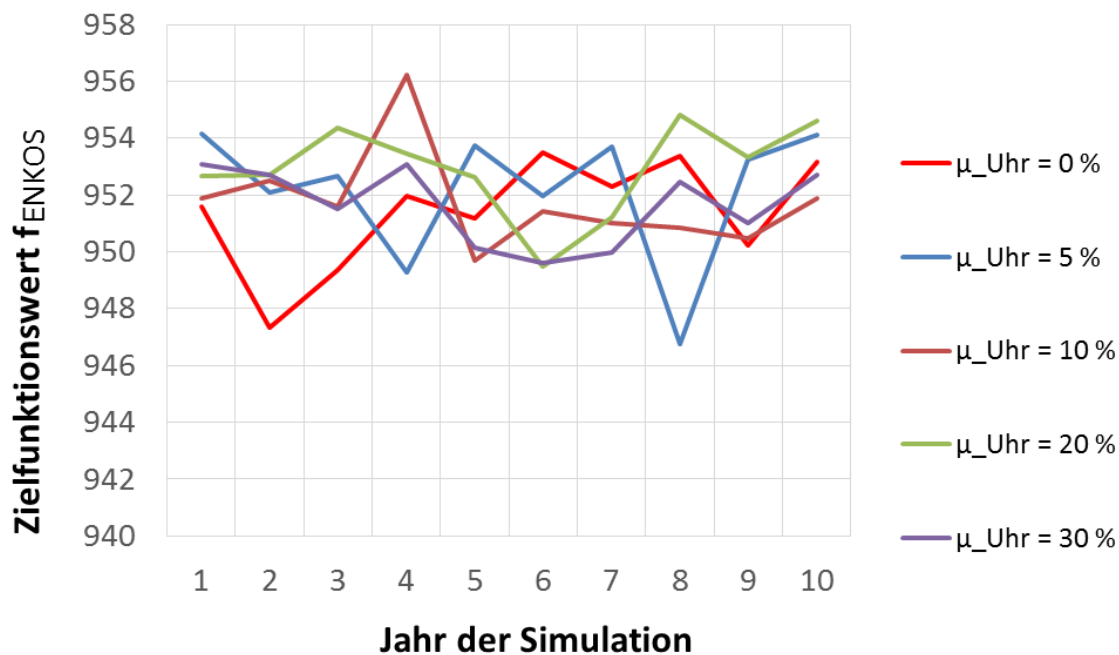


Abbildung 83. Simulation verschiedener Generalisierungsraten (Uhrzeit) für 10 Jahre.

In der Simulation, die in Abbildung 83 gezeigt ist, gibt es zunächst keine eindeutig dominierende Generalisierungsrate des Uhrzeitbereichs. Damit ist – im Gegensatz zur Lichtstärke – die Integration dieser Funktion durchaus berechtigt.

Bei einem Wert von $\mu_{Gen,Uhrzeit} = 20\%$ ergibt sich der größte Wert der Zielfunktion. Da die größte Differenz allerdings nur 1,5 ist und damit kleiner als die geringste Streuung, ist das Ergebnis mit Vorsicht zu genießen (vgl. Ab-

bildung 84). Der Einfluss dieses Parameters auf die Zielfunktion ist sehr gering und sollte zukünftig außer Acht gelassen werden.

Damit ist auch dieser Parameter nicht erheblich für die Steigerung des Werts der Zielfunktion f_{ENKOS} .

Generalisierungsrate $\mu_{\text{Gen,Uhrzeit}}$ in [%]	f_{ENKOS}	
	Mittelwert \pm Standardabweichung	Differenz zum besten Wert
0	951,4 \pm 1,9	1,5
5	952,2 \pm 2,4	0,7
10	951,8 \pm 1,7	1,2
15	951,7 \pm 2,6	1,2
20	952,9 \pm 1,6	0,0
25	952,8 \pm 1,6	0,1
30	951,6 \pm 1,3	1,3

Tabelle 21. Durchschnittswerte der Zielfunktion f_{ENKOS} in Abhängigkeit von der Generalisierungsrate der Uhrzeit $\mu_{\text{Gen,Uhrzeit}}$. Für die grün hinterlegte Generalisierungsrate von 20 % ergibt sich der größte Wert der Zielfunktion.

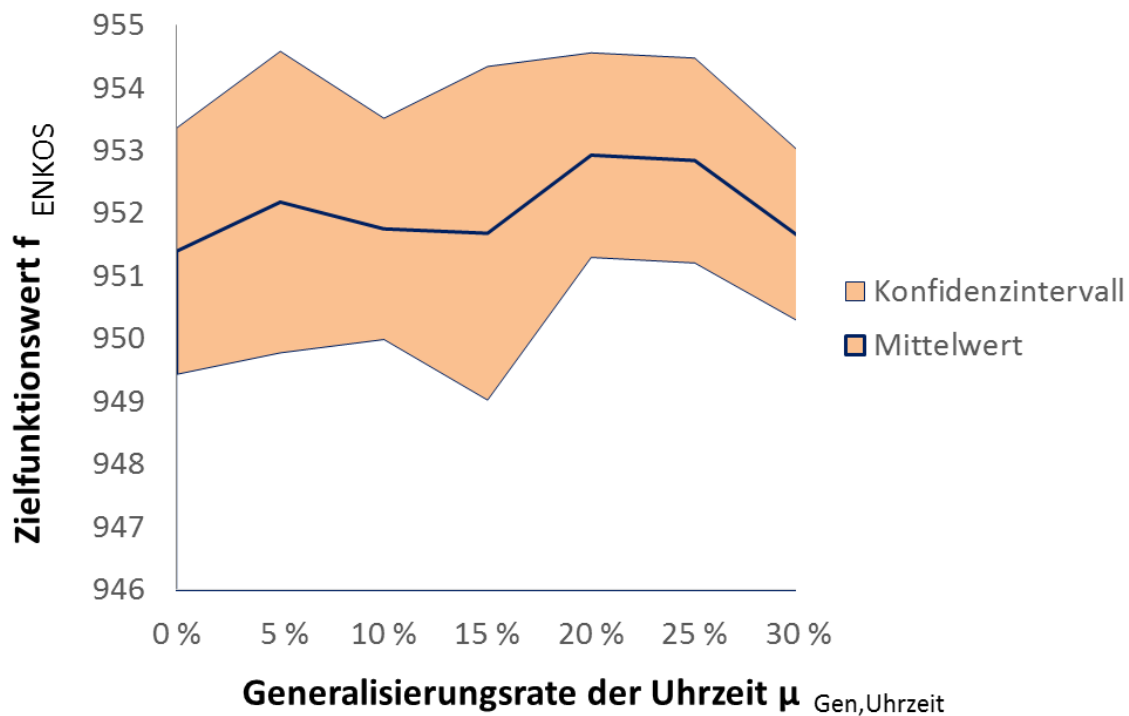


Abbildung 84. Darstellung von Mittelwert und Standardabweichung des Zielfunktionswerts in Abhängigkeit von der Generalisierungsrate des Uhrzeitbereichs. Es ist optisch gut ersichtlich, dass keine eindeutige Aussage getroffen werden kann, da die Standardabweichung deutlich größer ist als die Schwankungen des Mittelwerts.

7.6 Einfluss der Vergessensrate

Die Vergessensrate dient dazu, Regeln nach einer bestimmten Zeit zu verlieren. Das ist interessant, um menschliches Verhalten bezüglich abgelegter Gewohnheiten nachzubilden. Deshalb wird auch in ENKOS eine Vergessensrate als Wahrscheinlichkeit implementiert, eine Regel aus der Wissensbasis zu entfernen.

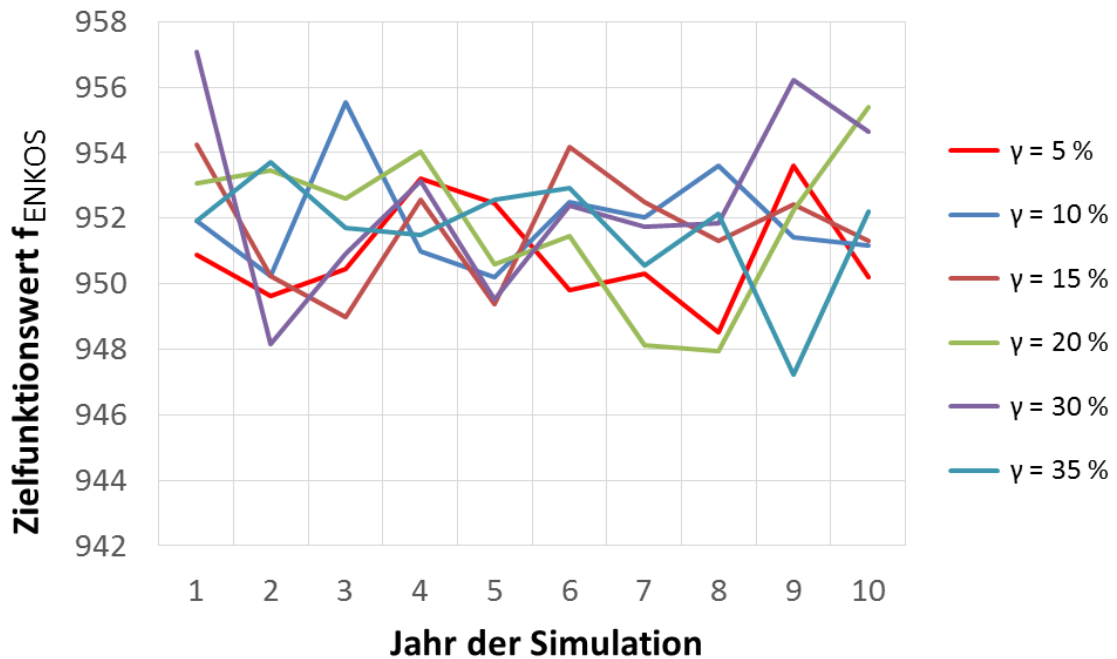


Abbildung 85. Darstellung verschiedener Vergessensraten von 10 simulierten Jahren.

In Abbildung 85 wird bereits deutlich, dass wiederum die Standardabweichung die Schwankungen der Mittelwerte übersteigt. Also ist ein Vergessen der Regeln nicht sinnvoll, sondern vielmehr die permanente Weiterentwicklung der Wissensbasis. Auch ist keine eindeutige Tendenz über die Lebensdauer der Wissensbasis hinweg erkennbar.

Da aus Tabelle 22, wie bereits bei vorherigen Parametern, ersichtlich wird, dass die Standardabweichung der Messwerte bereits mindestens so groß ist wie die Mittelwertschwankung selbst, hat die Vergessensrate keinen Einfluss auf die Maximierung der Zielfunktion (vgl. Abbildung 86). Dieser Parameter kann bezüglich der gegebenen Aufgabenstellung ignoriert werden.

Vergessensrate γ in [%]	f_{ENKOS}	
	Mittelwert \pm Standardabweichung	Differenz zum besten Wert
5	951,6 \pm 1,8	1,1
10	952,6 \pm 2,8	0,1
15	952,7 \pm 1,7	0,0
20	951,9 \pm 2,4	0,8
25	951,7 \pm 1,8	1,0
30	952,0 \pm 1,6	0,8
35	950,1 \pm 1,7	1,8

Tabelle 22. Durchschnittswerte der Zielfunktion f_{ENKOS} in Abhängigkeit von der Vergessensrate. Für die grün hinterlegte Vergessensrate von 15 % ergibt sich der größte Wert der Zielfunktion, wobei auch in diesem Fall die kleinste Standardabweichung bereits der Mittelwertschwankung entspricht.

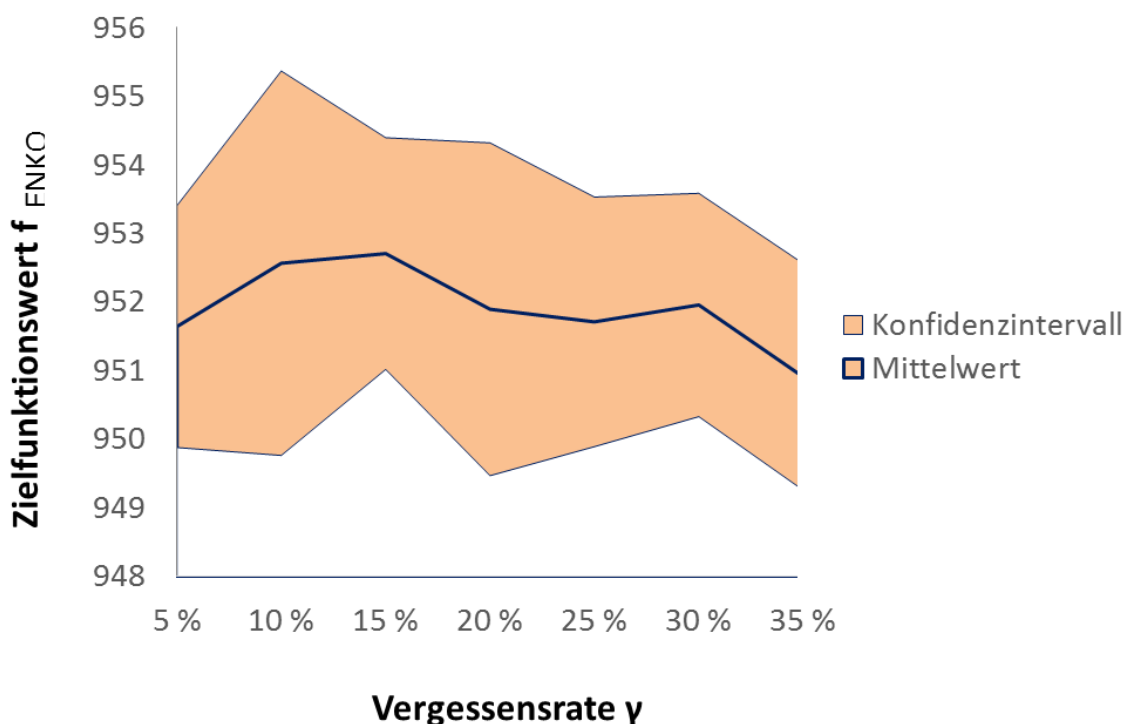


Abbildung 86. Darstellung des Zielfunktionswerts in Abhängigkeit der Vergessensrate.

7.7 Bewertung der Einflüsse

Zusammengefasst haben die Wahl der Mutations- sowie der Generalisierungsrate einen signifikanten Einfluss auf den Wert der Zielfunktion, alle anderen Parameter des LCS können ignoriert werden. Die Einflüsse können wie folgt für die entsprechend definierten Wertebereiche quantifiziert werden:

- Die **Mutationsrate** (μ_{Mut}) hat im Wertebereich von 1 ... 6 % eine Differenz beim Zielfunktionswert von **13,4** verursacht, während
- Die Generalisierungsrate (μ_{Gen}) im Wertebereich 1 ... 6 % sogar zu einer Differenz von **64,9** geführt hat.

Damit ist nachgewiesen, dass diese beiden Parameter wichtig sind für die Optimierung von ENKOS. Desweiteren ist wichtig, dass der Einsatz von ENKOS den Komfort für die Nutzer in der simulierten Wohnung gegenüber dem Nichteinsatz des Systems steigert.

Im Bereich der Methoden zur Energieminimierung sind die Erkennung der Nichtanwesenheit von Nutzern sowie die Substitution von Services aussichtsreiche Funktionen in ENKOS, wobei der Einfluss auf die Zielfunktion um den Faktor 10 reduziert ist.

8 Ausblick

Da in der vorliegenden Arbeit mit der Symbiose von Methoden der Künstlichen Intelligenz und dem großen Anwendungsgebiet des „Smart Home“ ein recht junges Forschungsgebiet näher untersucht wurde, sind in jeder Richtung weitere, intensive Untersuchungen denkbar. Davon sollen nachfolgend einige aussichtsreiche in Bezug auf konkrete Anwendungen aufgezeigt werden.

Aus psychologischer Sicht werden Menschen, deren Verhalten durch das System gelernt wird, und ENKOS selbst getrennt aufgefasst. Dabei dienen die Menschen als Lehrer, deren Verhalten durch ENKOS nachgeahmt soll. Tatsächlich ist es aber durchaus denkbar, dass auch eine Rückkopplung in die andere Richtung erfolgen kann. Das bedeutet, dass auch ENKOS die Menschen in seiner Umgebung durch gezielte Steuerstrategien beeinflussen kann. Beispielsweise ist eine Modellierung menschlichen Verhaltens durch das System denkbar, um etwa Energie zu sparen.

Desweiteren wurden Learning Classifier Systems als Werkzeug und mit analytischen Methoden kombiniert eingesetzt, um das Verhalten zu lernen. Dennoch existiert eine Vielzahl an weiteren möglichen Kombinationen, um menschliches Verhalten auf abstrakteren Ebenen zu lernen. Denkbar ist der verworfene Neuronale Netz-Ansatz, wenn er entsprechend erweitert und beispielsweise für die Darstellung einzelner Teile einer Regel verwendet wird. Auch eine Kombination mit Fuzzy-Methoden ist gut denkbar, wobei die Komplexität damit weiter gesteigert wird.

Im Rahmen der zunehmenden Vernetzung der Geräte, die die Menschen permanent in deren Alltag umgeben, sowie des Internets der Dinge ist ein Austausch von Verhaltensmustern genauso denkbar wie die Anwendung

einer Lernphase zur Programmierung scheinbar nicht lernfähiger Systeme. Als Resultat könnte ein Geschäftsmodell im Handel mit Nutzerprofilen für bisherige Smart Home-Systeme entstehen, die dann durch Verhaltensadaptation deutlich aufgewertet würden. Je nach Nutzergruppe wären weitere Generalisierungen möglich, auch der Vergleich verschiedener typischer Anwendungsgruppen oder gar die Einteilung in Reifegrade beim Energiesparen. Das wäre zum Beispiel sinnvoll, wenn mit zunehmendem Reifegrad der Nutzerwunsch permanent stärker Beachtung findet.

Desweiteren sollten Musterhaushalte gezielt mit Algorithmen ausgestattet werden, die die Tauglichkeit in der Praxis nachweisen. Hier wäre eine Verbindung mit den Haushalten von DigitalStrom sehr interessant. Denn dort ist bereits die Hardware in normale und repräsentative Wohnungen installiert und erste Ansätze der Erweiterung mit Algorithmen sind ebenfalls geplant. Das bedeutet, dass die Implementierung der dafür nötigen Schnittstelle ohnehin vorangetrieben wird und bereits die Nutzerakzeptanz gegeben ist.

Desweiteren ist ein deutscher Standardhaushalt erstellt worden, der in Normen für Smart Home-Systeme übernommen werden könnte. Damit ließe sich ein Wirkungsgrad bestehender Systeme messen und für neue fordern. Außerdem wird das Thema der Netzstabilität im Rahmen der Energiewende immer wichtiger. Auch hier könnte eine Weiterentwicklung von ENKOS erfolgen, um mit Hilfe einer zentralen Steuerung ein Demand-Side-Management so umzusetzen, damit Geräte in bestimmte Verbrauchsmodi gefahren werden, wenn beispielsweise keine Person anwesend ist. So könnte der Stromverbrauch auch künstlich hochgetrieben werden, wenn zu viel Wind weht und die Sonne gleichzeitig stark scheint. Dann wäre ein dynamischer Ausgleich der Netzzunter- oder -überlast denkbar.

Anlage A – Definition Kommunikationsprotokoll

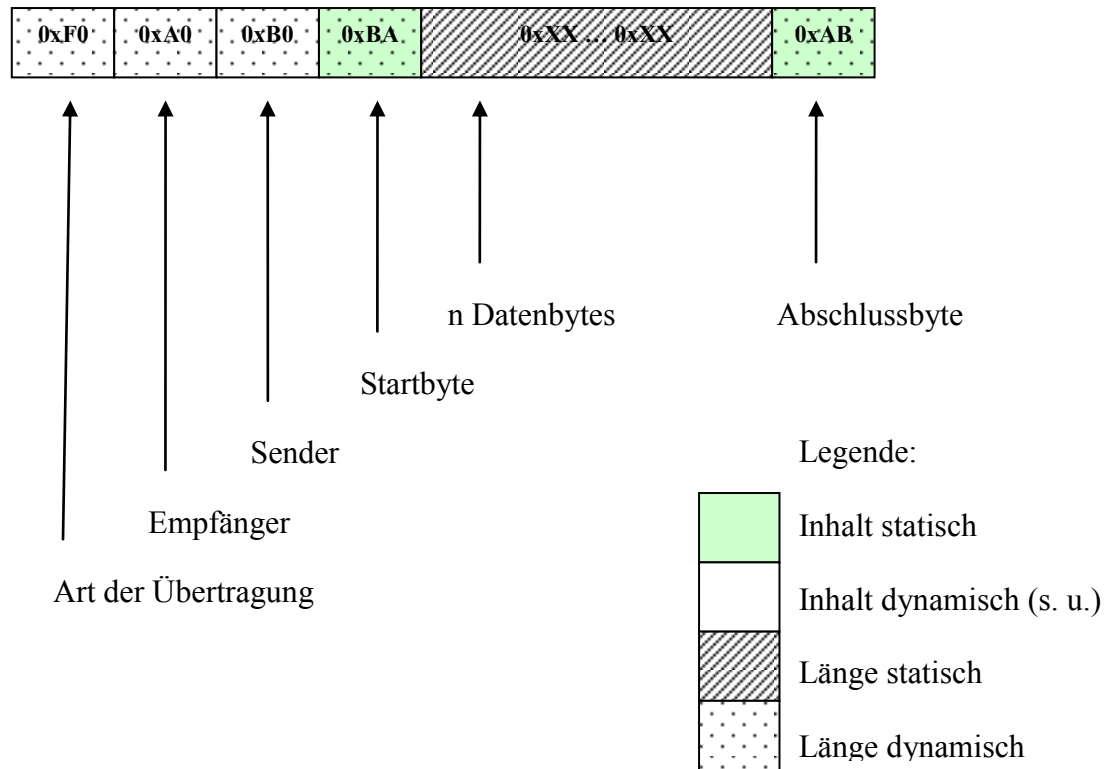


Abbildung 87. Ein Kommunikationswort zwischen dem PC und den Mikrocontrollern besteht aus einer Anzahl von Bytes, die sowohl die Art der Übertragung (Protokoll), den Absender und Empfänger, als auch die Datenbytes beinhaltet.

Anlage B – Kommunikationsdiagramm ENKOS/ Treiberapplikation

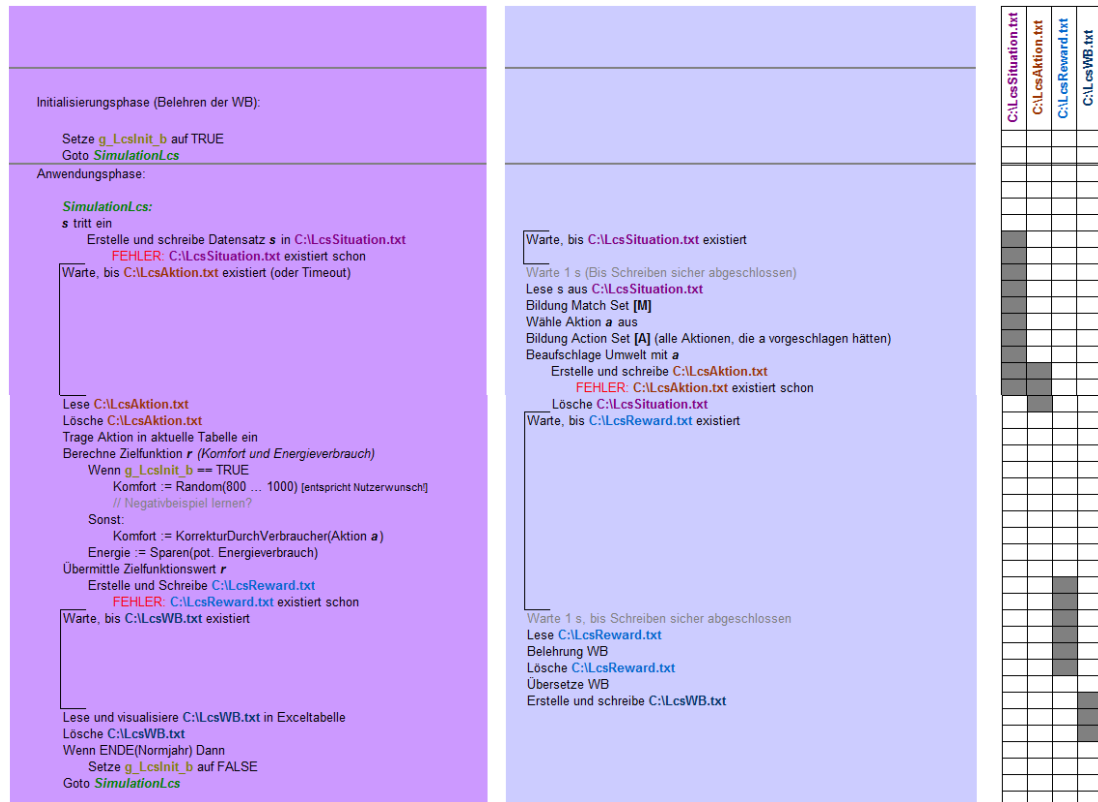


Abbildung 88. Kommunikationsablauf zwischen Treiber (Excel) und ENKOS (C++-Anwendung).

Anlage C – Neuronale Netze

Eine Methode, die häufig im Rahmen maschinellen Lernens eingesetzt wird, sind die Künstlichen Neuronalen Netze. Sie können als überwachte, bestärkende oder unüberwachte Methode genutzt werden. Dabei dienen das menschliche Gehirn und das Rückenmark, also biologische Neuronale Netze, als Vorbild. Eingesetzt wird diese Technik der Modellbildung vorrangig bei einer nicht linear beschreibbaren Umwelt, wie beispielsweise komplexen Regelungen, von denen verhältnismäßig wenige Informationen über interne Prozesse bekannt sind. Auch bei psychologischen Entscheidungsprozessen ist, wie später gezeigt wird, nur ein Bruchteil der Informationen nach außen sichtbar, sodass künstliche Neuronale Netze als Modellansatz prinzipiell auch hier in Frage kommen und deshalb hier näher erläutert werden.

Die ersten Neuronalen Netze wurden von Warren McCulloch und Walter Pitts 1943 entwickelt und waren eher wissenschaftlicher Natur, um zu erforschen, ob der Mensch mit diesem einfachen Modell seines Gehirns überhaupt einfache turingberechenbare Funktionen berechnen kann [83]. Das war der Fall, weil durch eine schrittweise, mathematisch exakt erfassbare Abfolge von einfachen Algorithmen ein aktueller Zustand eindeutig in einen Zielzustand überführt werden konnte.

In der Informatik hat man Künstliche Neuronale Netze weiterentwickelt, nicht mehr, um das menschliche Denken zu verstehen (wie im psychologischen Teilforschungsgebiet), sondern weil vielmehr die Vorteile dieser hyperparallelen Informationsverarbeitung sichtbar wurden [124]. Gerade die enormen Fortschritte bei der Rechentechnik ließen die Vorteile dieser Art der Datenverarbeitung hervorheben.

Der Aufbau einer Zelle leitet sich jedoch nach wie vor aus dem biologischen Vorbild, das in Abbildung 89 gezeigt ist, ab. Es erhält eine Nervenzelle (Zellkörper, oder Neuron) über seine Dendriten (Zelleingänge) bestimmte Informationen in Form elektrischer Kleinströme im Mikroamperebereich.

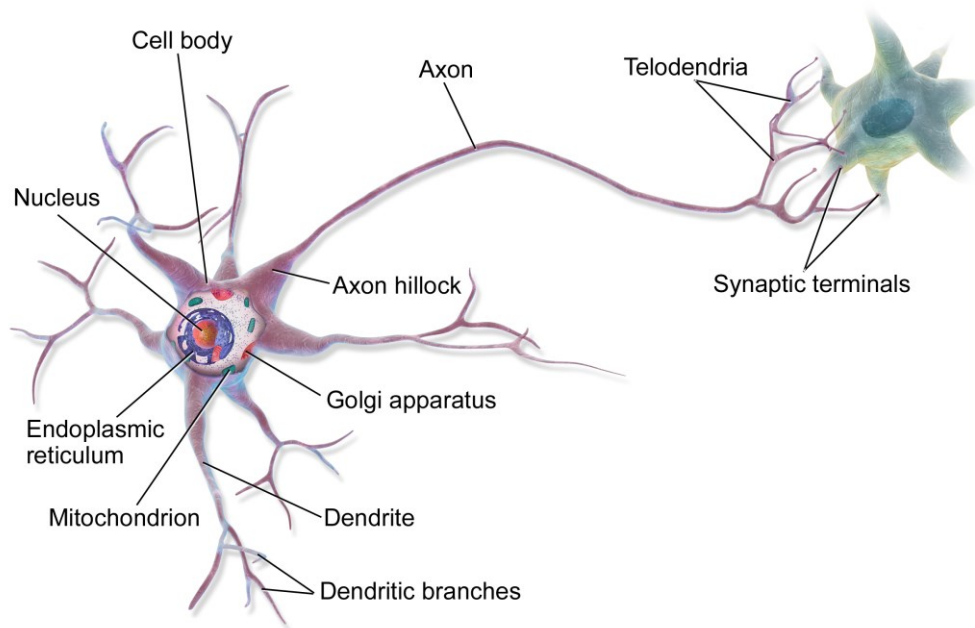


Abbildung 89. Biologisches Vorbild eines künstlichen Neuronalen Netzes [67].

Die zu einem gesamten Signal über alle Eingänge aufsummierte Stromstärke fließt in die Entscheidung darüber, welche Ausgangsinformation über das Axon (Nervenfaser) generiert wird, direkt ein. Diese Entscheidung wiederum wird durch eine Verbindung an durchschnittlich 14.000 andere Neuronen verteilt. Maximal sind im menschlichen Gehirn jeweils 200.000 Ein- und Ausgänge pro Zelle möglich [110].

Die Nachbildung dieser biologischen Zelle ist in Abbildung 90 gezeigt. Auch hier werden im ersten Schritt die j mit $w_1 \dots w_j$ gewichteten Eingangswerte aufsummiert (69) und das Resultat anschließend mit Hilfe einer

Aktivierungsfunktion sowie einer Ausgangsfunktion als Neuronenausgang y_σ berechnet.

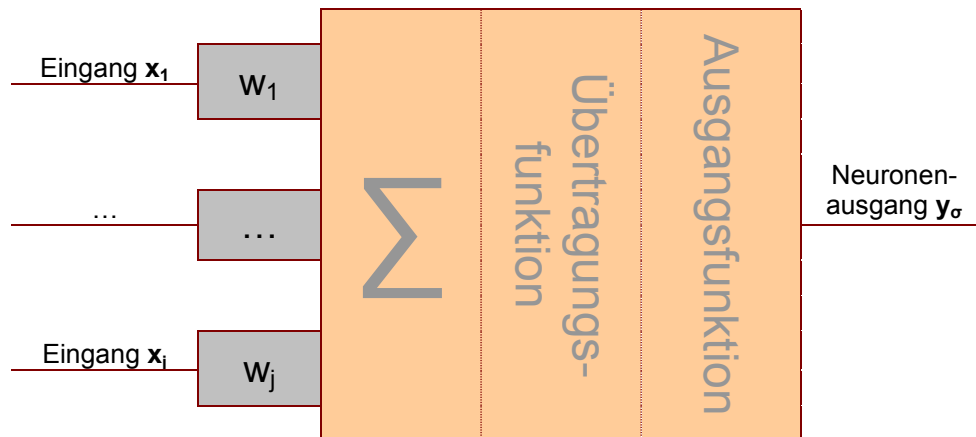


Abbildung 90. Nachbildung der biologischen Zelle im Neuronalen Netz (eigene Darstellung nach [103]).

$$y = \sum_{k=1}^j w_k x_k (+w_0) \quad (69)$$

Typischerweise wird eine sigmoide Aktivierungsfunktion gewählt. Aus dieser Klasse von mathematischen Funktionen werden in der Regel entweder die logistische Funktion (70a) oder der Tangens Hyperbolicus (70b) ausgewählt (vgl. Abbildung 91). Beide Funktionen sind an allen Stellen stetig und differenzierbar, was wichtig für die Anwendung des Backpropagationalgorithmus ist.

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (70a)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (70b)$$

Die Wahl der Funktion richtet sich nach der Aufgabenstellung, für viele Probleme jedoch sind die genannten Funktionen zielführend.

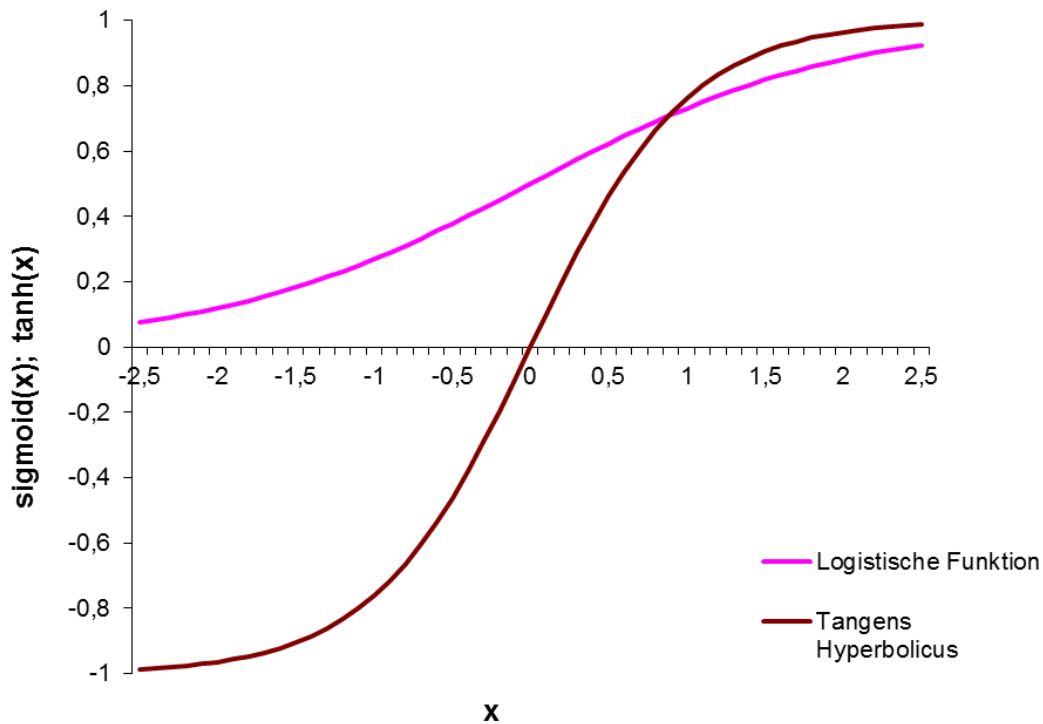


Abbildung 91. Visualisierung der Sigmoidfunktionen „Logistische Funktion“ und „Tangens Hyperbolicus“.

Bei der Ausgangsfunktion wird normalerweise die Identitätsfunktion, wie in (71) gezeigt, benutzt [103], die in den meisten Anwendungen die besten Resultate erzielt und einfach zu handhaben ist, während wenige Literaturstellen für spezielle Anwendungen eine Schwellenwertfunktion empfehlen [101]. Sowohl die Gewichtungs- als auch die Aktivierungsfunktion sind in der Regel fest im Netzwerk als Parameter verankert [66], sodass keine Änderung im laufenden Lernprozess erfolgt.

$$y = x \quad (71)$$

Die etwa 100 Milliarden Neuronen im menschlichen Gehirn machen dieses so leistungsfähig, dass bis heute eine rechentechnische Nachbildung noch nicht möglich ist. Das erste Modell eines künstlichen, und damit rechnerbasierten Netzwerks aus Neuronen geht auf Frank Rosenblatt zurück und wird als „Perzeptron“ bezeichnet [36], das die Grundlage vieler heute verwendeter Topologien bildet.

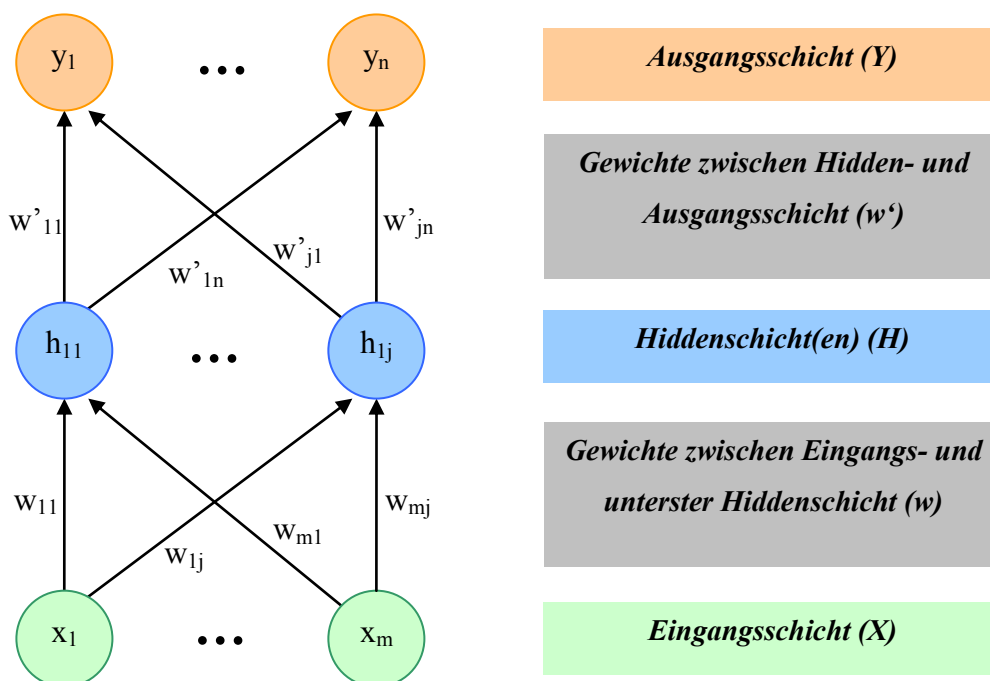


Abbildung 92. Ein Perzeptron [120].

Dieses ist in Abbildung 92 mit einer Hiddenschicht dargestellt und besteht damit aus m Eingabeneuronen ($x_1 \dots x_m$), die die Eingangsschicht X bilden. Diese sind mit dem Neuronen der Hiddenschicht H (durch die Integration von Hiddenschichten wird das Perzeptron zum Multi-Layer-Perzeptron, MLP [97]), die nach außen nicht sichtbar ist, über die jeweiligen Gewichte

$w_{11} \dots w_{mj}$ verbunden, wobei der erste Index die Nummer des Eingangs- und der zweite Index die Nummer des Hiddenneurons abbildet.

Das menschliche Lernen selbst spiegelt sich im biologischen neuronalen Netz in der Veränderung der Synapsen zwischen zwei Neuronen wieder, sodass in den Werten dieser Verbindungen das eigentliche Wissen gespeichert ist. An diesen werden aufgrund einer biochemischen Reaktion die Neurotransmitter freigesetzt, die durch die Axone transportiert werden. Diese Neurotransmitter können dabei hemmend (inhibitorisch) oder verstärkend (exzitatorisch) sein, also die elektrische Ladung zu größer oder kleiner Geltung im aufnehmenden Neuron kommen lassen. Außerdem wird die Größe der Synapsen, die Verbindungsstellen zwischen Axonen und Neuronen, während des Lernprozesses verändert sowie Axone generiert (Herstellung eines Zusammenhangs) oder zerstört (kein Zusammenhang im menschlichen Denken) [110].

Über die Gewichte $w'_{11} \dots w'_{jn}$ sind die Hiddenneuronen wiederum mit den n Neuronen der Ausgangsschicht Y , und zwar $y_1 \dots y_n$ verbunden. Um später einfacher Zusammenhänge bei den Lernverfahren zu erläutern, wird die Gesamtheit aller Gewichte als Matrix \mathbf{W} definiert (als Beispiel für ein Netzwerk ohne Hiddenschichten lt. Abbildung 92 in (72) gezeigt), wobei in diesem Fall der Betrag $\|\mathbf{W}\|$ als Anzahl der skalaren Elemente in \mathbf{W} bezeichnet wird [105].

$$\mathbf{W} = \begin{bmatrix} w_{11} & \dots & w_{mj} \\ w'_{11} & \dots & w'_{jn} \end{bmatrix} \quad (72)$$

Damit ist in Abbildung 92, sofern m, j und n definiert sind, eine vollständige Topologie für ein künstliches Neuronales Netz gezeigt. Eingangs- und Aus-

gangsschicht sind dabei die Schnittstellen zur Anwendung, sodass meistens die Eingangsneuronen Sensorinformationen oder Anwenderaktionen aufnehmen. Die Ausgangsneuronen stellen oftmals Steuersignale oder Systemstatus dar.

Die Zusammenfassung der Neuronen zu Schichten wird in den meisten Topologien vorgenommen, wobei Neuronen einer Schicht gleichartige Aufgaben hinsichtlich der mathematischen Interpretation des Netzes übernehmen. Das Hinzufügen bzw. Weglassen ganzer Schichten entspricht der Modifikation der Ordnung des gesamten Modells [46] [86].

Die Gewichte als eigentliche Träger der Modellinformation werden während des Lernprozesses durch einen entsprechenden Lernalgorithmus, in der Regel Backpropagation [110] [120] [66], adaptiert, indem der durch das Netz generierte Ausgangsvektor $y_a(t) = [y_1(t), \dots, y_n(t)]$ mit dem Sollvektor $\hat{y}_a(t) = [\hat{y}_1(t), \dots, \hat{y}_n(t)]$ während jedes einzelnen Lernschritts verglichen wird. Dabei kommt es (im nicht perfekten Modell) zu einer Differenz, dem Fehler $E_{\text{Netz}}(t)$, der als skalarer Wert den Lernfortschritt dokumentiert und normalerweise gegen einen Grenzwert E_{Netz}^* konvergiert. Dieser Fehler soll für alle vorliegenden Datensätze $d = [1 \dots N]$ minimiert werden. In (73) ist die Berechnung anhand der Fehlerquadratsumme über alle einzelnen Ausgabe-neuronen und alle vorliegenden Datensätze gezeigt (der Faktor 1/2 erleichtert lediglich die Ableitung der Funktion, hat aber auf das Ergebnis an sich keinen Einfluss). Andere Definitionen von Fehlern sind dabei auch möglich, je nach Anwendung.

$$E_{\text{Netz}}(t) = \frac{1}{2} \sum_d (y_a(t) - \hat{y}_a(t))^2 \quad (73)$$

Je nach Problemstellung kann ein Netzwerk vorwärts- oder rückwärtsgerichtet Informationen verarbeiten. Während die Vorwärtsrichtung (Feed-Forward, FF) einem azyklischen³⁰ Graphen entspricht [147] und beispielsweise für Klassifikationsprobleme eingesetzt werden kann, eignen sich rückwärtsgerichtete Netze (Feed-Backward, FB) eher zyklischen Problemstellungen.

Weiterhin wird unterschieden zwischen vollverketteten (alle Neuronen einer Schicht sind mit allen Neuronen der darüber- bzw. darunterliegenden Schicht durch Gewichte verbunden) oder teilverketteten Netzwerken, wobei eine teilweise Verkettung auch ein Sonderfall darstellen kann, wenn einige Gewichte während des Lernens unter einen bestimmten Wert fallen und damit vernachlässigbar sind.

Während der Lernphase selbst werden dem Netzwerk eine Anzahl von Trainingsdatensätzen N , die aus allen verfügbaren Datensätzen $N+M$ ausgewählt werden, als Trainingsmenge N nacheinander präsentiert. Erfahrungsgemäß werden am besten zwei Drittel zufällig ausgewählt, wobei wichtig ist, dass die Datensätze ein repräsentatives Abbild des Zustandsraums sind. Dabei wird das Netz mit jedem Eingangsdatensatz l aus N beaufschlagt und die Werte der Ausgabeneuronen berechnet (zur Initialisierung werden zufällige Gewichte benutzt, womit die Ausgabe $y_a(l)$ des Netzes sehr vom Sollwert abweichen wird). Anschließend wird der Netzwerkfehler über alle zur Verfügung stehenden Sätze mit Hilfe von (73) berechnet.

Als letzter Schritt eines jeden Trainingsdatensatzes wird der berechnete Fehler $E_{\text{Netz}}(t)$ „rückpropagiert“, um die Gewichte des Netzes so anzupassen,

³⁰ In einem azyklischen Graphen fließen die Informationen nicht zwangsläufig von links nach rechts, sondern auch ggf. in die andere Richtung.

dass mit dem Netz eine bessere Abbildung des gelernten Datensatzes erfolgt. Dabei wird davon ausgegangen, dass die Fehlerfunktion jedes Neurons im Netzwerk durch die Aktivierungsfunktion abhängig von dessen Eingangswerten ausgedrückt werden kann und die Fehlerfunktion des gesamten Netzwerks als Funktion abhängig von allen Neuronen. Diese besitzt ein Minimum, bei dem eine bestimmte Netzkonfiguration zu einer minimalen Abweichung zwischen Soll- und Istausgangswerten führt. Das mathematisch mittels der ersten partiellen Ableitungen der Netzfehlerfunktion $E_{\text{Netz}}(t)$ nach allen Netzgewichten w_{ij} , wie in (74a) dargestellt ist, ermittelt werden [104]. Dabei ist y_σ der Ausgabewert des Neurons und δ_{ij} der Propagierungsfehler, also die Differenz des aktuellen zum gewünschten Ausgabewert des Neurons.

$$\frac{\partial E_{\text{Netz}}(t)}{\partial w_{ij}} = \frac{\partial E_{\text{Netz}}(t)}{\partial y_\sigma} \frac{\partial y_\sigma}{\partial w_{ij}} = y_\sigma \delta_{ij} \quad (74a)$$

$$\Delta w_{ij}(t+1) = (1-\eta)\Delta w_{ij}(t) + \eta \delta_{ij} \quad (74b)$$

Anschließend kann mithilfe einer Lernrate η in (74b) das Gradientenabstiegsverfahren angewandt werden, um die Gewichte w_{ij} so anzupassen, um den Netzfehler $E_{\text{Netz}}(t)$ zu minimieren. Die Lernrate η selbst wird dabei in der Regel zuerst im Bereich $\eta = 0,7$, mit fallendem Propagierungsfehlern δ_{ij} jedoch immer kleiner bis in den Bereich $\eta = 0,05$ gewählt, um Oszillationen nahe des Ziels zu vermeiden.

Da der Fehler zuerst für die Ausgangsneuronen bekannt ist, wird dieser Algorithmus auch zuerst für diese angewandt. Danach arbeitet er sich von den Hiddenneuronen (ausgangsseitig) bis hin zur Eingangsschicht rückwärts

durch, bis alle Gewichte entsprechend bearbeitet wurden, um den Netzfehler zu minimieren. Am Anfang der Lernphase werden sich die Gewichtswerte noch signifikant ändern, wenn diese Änderung nicht mehr spürbar ist (also unter einen Schwellwert liegt), wird die Lernphase als abgeschlossen betrachtet.

Aufgrund der biologischen Ähnlichkeit sowie der Möglichkeit der stark nichtlinearen Abbildung von Funktionen können künstliche Neuronale Netze als Vorteile hinsichtlich der Problemstellung dieser Arbeit selbstständig und adaptiv menschliche Verhaltensweisen, auch aus unvollständigen Informationen lernen. Außerdem können sie, ähnlich wie das menschliche Gehirn arbeitet, Wissen generalisieren sowie ähnliche Eingangsmuster miteinander assoziieren. Sie arbeiten robust gegenüber gewissen Abweichungen bei der Anfangstopologie und bilden, ähnliche wie menschliches Verhalten, sehr komplexe und miteinander in Wechselwirkung stehende Zusammenhänge ab.

Der größte Nachteil ist, dass das Wissen, das in einem Netz steckt, sehr schwer entschlüsselt werden kann. Nach einem abgeschlossenen Lernvorgang muss, wenn das überhaupt mit vertretbaren Ressourcen noch möglich ist, eine langwierige Auswertung der Gewichtswerte vorgenommen werden. Damit ist meist nur eine vergleichende Analyse oder ein Reverse Engineering-Ansatz möglich. Außerdem kann leicht, wenn das Netzwerk zu groß gewählt wurde (d. h. zu viele Hiddenschichten), eine zu große Spezialisierung (Overfitting) bzw. bei einem zu kleinen Netzwerk zu große Generalisierung vorkommen (Underfitting). Daneben ist eine sehr große Anzahl von repräsentativen Datensätzen nötig, in eigenen Experimenten mehrere zehntausend die manchmal nur schwer gewonnen werden können. Zudem ist ei-

ne Online-Verarbeitung von Wissen erst bei einem sehr gut trainierten Netz möglich, zumal meist die Lern- von der Arbeitsphase getrennt werden.

Netzwerktypen

Generell können Neuronale Netzwerke in verschiedene Netztypen gegliedert werden [103], wobei für jeden dieser Typen wiederum bestimmte verschiedene Lernregeln, Netzwerktopologien, usw. verwendet werden können. In dieser Arbeit sollen vier Typen in Bezug auf die vorliegende Problemstellung betrachtet werden. Weiter oben in diesem Abschnitt wurde bereits der einfachste Netzwerktyp dargestellt, um die allgemeinen Gesetzmäßigkeiten zu zeigen.

Das ***musterassoziative Netz*** (Pattern Associator) ist lediglich zweischichtig, verzichtet also gänzlich auf Hiddenneuronen, wie in Abbildung 93 dargestellt. Wie der Name bereits aussagt, wird dieser Netzwerktyp aus der Gruppe der Feedforward-Netze in erster Linie zur Mustererkennung genutzt, wobei auch Signalreparaturen oder Generalisierungen von Mustern durchgeführt werden können. Die Netze sind in der Regel vollvernetzt und vorwärtsverbunden.

Aufgrund der fehlenden Hiddenschichten eignen sie sich zum Belehren des Netzes die recht frühen Methoden Deltaregel oder Hebb-Regel. Die Hebb-Regel entstammt dabei dem biologischen Vorbild Gehirn gemäß dem Motto „What fires together, wires together“, was soviel bedeutet, als dass Gewichte zwischen Neuronen, die gemeinsam bei einem bestimmten Reiz aktiv sind, miteinander stärker verbunden (assoziiert) werden [62].

$$\Delta w_{ij} = \varepsilon \cdot y_i \cdot y_j \quad (75)$$

Das Hebbsche Lernen (75) kann im Rahmen von überwachten, bestärkenden oder unüberwachten Lernmethoden eingesetzt werden.

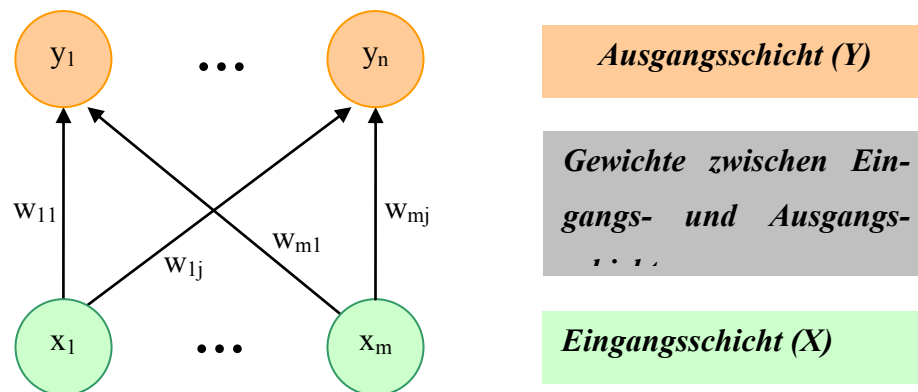


Abbildung 93. Ein musterassoziatives Netz (eigene Darstellung).

Die Deltaregel als alternative Lernregel für diesen Netzwerktyp hingegen ist nur als überwachtes Lernen denkbar, da hier der erwartete Ausgangswert \hat{y}_i des Neurons i mit dessen wirklichen Ausgangswert y_i verglichen, mit dem Lernfaktor ε gewichtet und mit dem aktuellen Ausgangswert y_i verrechnet wird. Das ist in (76) gezeigt.

$$\Delta w_{ij} = \varepsilon \cdot y_j \cdot \delta_j = \varepsilon \cdot y_j \cdot (\hat{y}_j - y_j) \quad (76)$$

Leicht zu erkennen ist, dass das Gewicht zwischen Neuron i und j in Richtung des Fehlers (Abweichung zwischen Ausgabe- und Sollwert) verändert wird. Der Lernparameter kann bei beiden Lernverfahren entweder fest vorgegeben werden oder während des Lernprozesses in Abhängigkeit vom noch

vorhandenen Fehler angepasst werden. Aufgrund der fehlenden Komplexität sind diese Netzwerke nicht für den Verhaltenserwerb geeignet.

Sehr damit verwandt sind *kompetitive Netze*. Der Aufbau ist prinzipiell der gleiche, nur wird als Lernregel ausschließlich Wettbewerbslernen benutzt, wobei die Topologie des Problems dem unüberwachten Lernen entspricht. Dabei werden die Gewichte anhand (77) aktualisiert und damit dem beaufschlagten Eingangsmuster um den Lernfaktor ε ähnlicher gemacht [101].

$$\Delta w_{ij} = \varepsilon(a_j - w_{ij}) \quad (77)$$

Die Anwendungsgebiete liegen dabei ebenfalls vorrangig in der Musterklassifikation, nicht jedoch zur Verhaltensadaption, doch im Gegensatz zum musterassoziativen Netz als unüberwachtes Lernverfahren [84]. Desweiteren eignet es sich durch sein spezielles Lernverfahren zur Reduzierung von Redundanzen in Signalketten, die wiederum durch weitere Instanzen maschinellen Lernens verarbeitet werden können.

Eine weitere Modifikation, ebenfalls als Methode unüberwachten Lernens eingesetzt, stellen die *Kohonennetze* dar. Der Unterschied ist die mehrdimensionale Struktur der Ausgangsschicht. Bei der Funktionsapproximation oder der grafischen Transformation im Bereich Computergrafik liegen die Stärken dieser Netze.

Eine typische zweidimensionale Ausgabeschicht ist in Abbildung 94 gezeigt und geht im Wesentlichen auf Teuvo Kohonen [64] zurück. Sowohl die Anzahl der Lernschritte als auch die Lernrate und die Dimension des Netzes (wie beispielsweise die Anzahl der Ausgabeneuronen) an sich haben entscheidenden Einfluss auf den Erfolg der Modellbildungsaufgabe. Aus diesem Grund bedarf es viel Erfahrung, ein Kohonennetz korrekt zu spezifizie-

ren, der technisch mögliche Wirkungsgrad bei der gegebenen Problemstellung ist jedoch nicht groß genug.

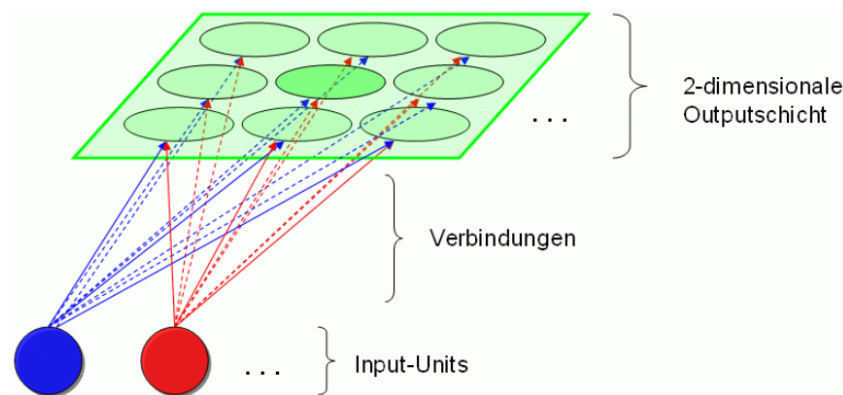


Abbildung 94. Darstellung Kohonennetz mit zweidimensionaler Ausgabeschicht [8].

Liegt eine Problemstellung vor, bei der die Umwelt überwiegend durch zyklische Abläufe oder Gedächtnisanwendungen beschrieben werden kann, werden in der Regel **rekurrente Netze** als Basis für das Modell benutzt. Diese sind durch die Anwesenheit von mindestens einer Hiddenschicht sowie verschiedene Rückkopplungen charakterisiert, die in Abbildung 95 visualisiert sind. Dabei spricht man von einem vollständigen Netz, wenn jedes Neuron mit jedem verbunden ist. Da die Übersichtlichkeit der Netztopologie mit zunehmender Komplexität sehr schnell abnimmt, werden normalerweise Verbindungen zwischen den Neuronen nur noch als Matrix zusammengefasst.

Vor allem zur Vorhersage menschlichen Verhaltens werden diese Netze benutzt. Drei bekannte Untertypen sind dabei das von Jeffrey L. Elman [31] entwickelte Simple Recurrent Network sowie Jordan- und Elman-Netze, die hier nur erwähnt werden sollen.

Die bekannten Probleme des Gradientenabstiegsverfahrens spielen hier eine Rolle. Zum einen wird konvergiert dieses Verfahren allzu oft in lokalen Mi-

nima, da es stets den Weg ins nächste Tal folgt. Hierbei ist eine Kombination mit evolutionären Algorithmen (parallel oder sequentiell) möglich, um mehrere Startpunkte für den Gradientenabstieg festzulegen. Außerdem hat das Verfahren mit flachen Plateaus Probleme, weil kein signifikanter Gradient mehr ermittelbar ist und somit das Verfahren irgendwann nicht mehr vorankommt.

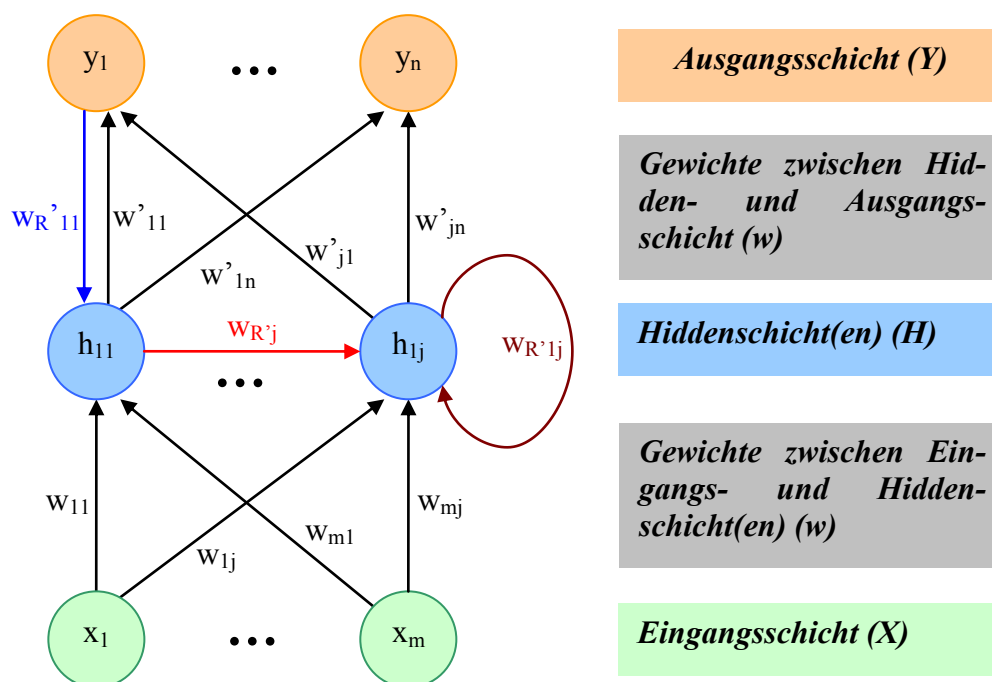


Abbildung 95. Ein rekurrentes Netz mit verschiedenen Rückkopplungen (eigene Darstellung). Die blaue Verbindung $w_{R'11}$ ist dabei die indirekte Rückkopplung von Neuron y_1 zu h_{11} , während die braune Verbindung $w_{R'lj}$ eine direkte Rückkopplung darstellt. Die rote Rückkopplung $w_{R'j}$ hingegen ist eine seitliche Rückkopplung innerhalb einer Schicht.

Rekurrente Netze werden durch den Backpropagationalgorithmus belehrt, dessen Ansatz ein Netzausgabefehler $E_{\text{Netz}}(t)$ in Abhängigkeit der gewählten Gewichte \mathbf{W} (und \mathbf{W}') für den aktuellen Lerndatensatz bildet. Diesen Fehler kann man sich als $\|\mathbf{W}\|$ -dimensionales Gebirge vorstellen, dessen (globales)

Minimum gleichzeitig den kleinsten Fehler des Netzes abbildet. Deshalb wird für den aktuellen Lerndatensatz per Gradientenabstiegsverfahren versucht, das Fehlergebirge in Richtung Minimum zu durchlaufen.

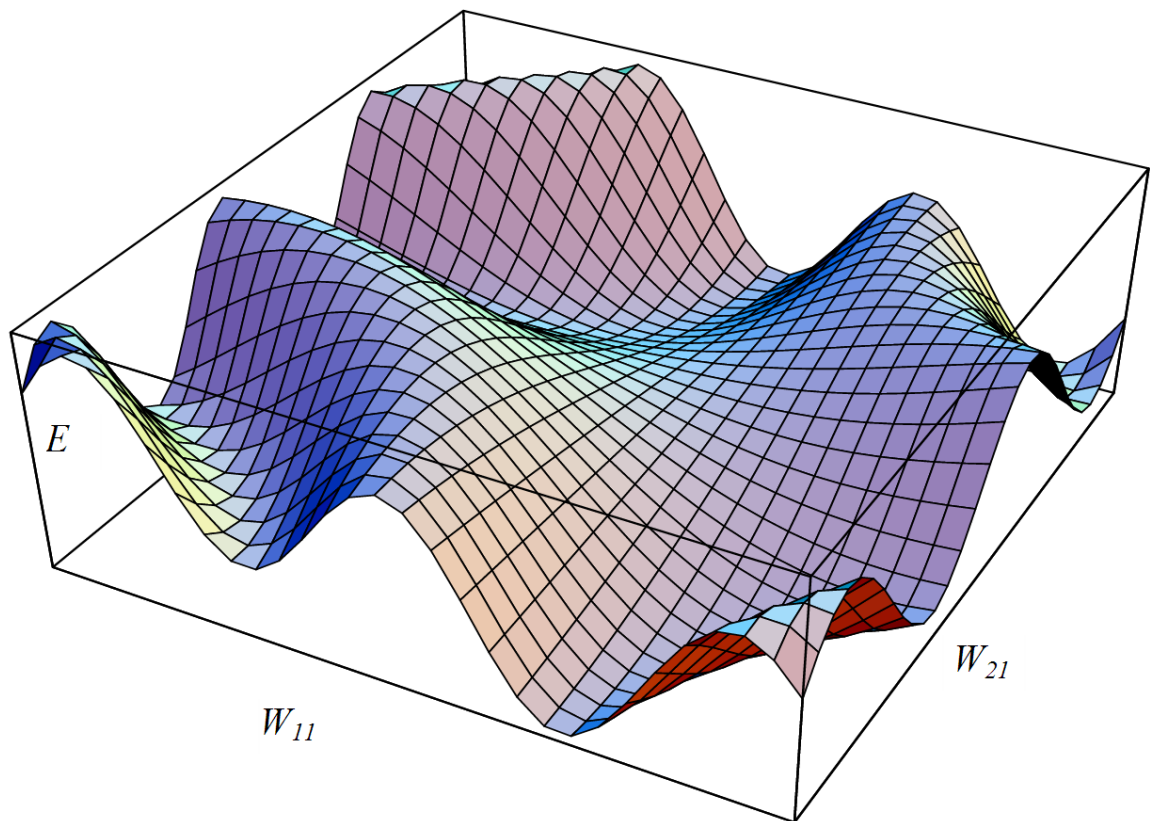


Abbildung 96. Ein typisches Fehlergebirge nach [97], wobei $E_{\text{Netz}}(t)$ in Abhängigkeit von zwei Gewichten w_{11} und w_{21} dargestellt ist, die Anzahl der Elemente in W somit lediglich 2 betragen würde.

Häufig wird eine „Trägheit“ des vorherigen Schrittes in Form eines Momentumterms bei der Wahl der neuen Richtung berücksichtigt [80] oder die Lernrate exponentiell reduziert, wie es in der simulierten Abkühlung nach dem Vorbild der Natur häufig beobachtet wird [130].

Literaturverzeichnis

- [1] Aizo, Digitalstrom, Internetquelle: www.aizo.com, Aufruf vom 24.2.2014.
- [2] Aizo, Log-Datei „system-event.log“ (als E-Mail von Fr. Schröter) vom 8.7.2014.
- [3] Aizo, Produktkatalog „DigitalStrom“, 2014.
- [4] Alpaydin, E., Introduction to Machine Learning (2. Auflage), MIT Press, Cambridge, 2010.
- [5] Badlani, A. a., Smart Home System Design based on Artificial. In: Proceedings of the World Congress on Engineering and Computer Science 2011 Vol I., San Francisco, USA, 2011.
- [6] Baron, R. S., Kerr, N. L., Miller N., Group process, group decision, group action, Pacific Grove, CA: Brooks/ Cole, 1992.
- [7] BDEW Bundesverband der Energie- und Wasserwirtschaft e.V. BDEW-Strompreisanalyse Mai 2013 - Haushalte und Industrie, 2013.
- [8] Beck, F., Internetquelle: www.neuronalesnetz.de, Abruf am 30.1.2014
- [9] Becker, J., Kugeler, M., Rosemann, M., Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung. Springer-Verlag Heidelberg Berlin, Münster, 2008.
- [10] Bellmann, R., Dynamic Programming, Princeton University Press, USA, 1957.
- [11] Bernadó-Mansilla, E., Garrell-Guiu, J., M., Accuracy-Based Learning Classifier Systems: Models, Analysis and Applications to Classification Tasks, In: Evolutionary Computation, IEEE Press, Piscataway, 2006.
- [12] Bernoulli, D., Exposition of a new theory on the measurement of risk, aus Econometrica, JSTOR, 1954.
- [13] Betsch, T., Funke, J., Plessner, H., Allgemeine Psychologie für Bachelor: Denken - Urteilen, Entscheiden, Problemlösen, Springer-Verlag Heidelberg Berlin, 2010.
- [14] Billerbeck, J. D., Der mühsame Weg zum smarten Zuhause, VDI nachrichten vom 15.8.2014, VDI Verlag GmbH, Düsseldorf, 2014.
- [15] Birkner, P., Der urbane Raum als Zentrum der Energiewende-Projekte und Konzepte aus der Praxis, Energy Ecosystems Conference, Leipzig, 2014.

- [16] Brooks, R. A., Maes, P., Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, MIT Press, USA, 1994.
- [17] Browne, W. N. L., Balancing Specificity and Generality in a Panmictic-Based Rule-Discovery Learning Classifier System, Lecture Notes in Computer Science Volume 2661, , pp 1-19, Springer-Verlag Berlin Heidelberg, 2003.
- [18] Buchanan, J. M., Game theory, mathematics, and economics, In: Journal of Economic Methodology, 8. Jg., Nr. 1, S. 27-32, Routledge, London, UK, 2001.
- [19] Butz, A. M., David, B., Goldberg, E., Stolzmann, C. W., The anticipatory classifier system and genetic generalization, Kluwer Academic Publishers, Alphen aan den Rijn, NL, 2002.
- [20] Butz, M. V., Goldberg, D. E., Bounding the population size in XCS to ensure reproductive opportunities, In: Proceedings of the Fifth Genetic and Evolutionary Computation Conference (GECCO-2003), pp. 1844-1856, Springer-Verlag Berlin Heidelberg, 2003.
- [21] Butz, M., Sastry, K., Goldberg, D., Tournament selection: stable fitness pressure in XCS, In: Proceedings of the Genetic and Evolutionary Computation Conference, Lecture Notes in Computer Science, pp. 1857-1869, Springer-Verlag Berlin Heidelberg, 2003.
- [22] Cichosz, P., Mulawka, J. J., Faster Temporal Credit Assignment in Learning Classifier Systems, In: Proceedings of the first Polish Conference on Evolutionary Algorithms, Warsaw University of Technology, 1996.
- [23] Cover, T. M., Thomas, J. A., Elements of Information Theory, In: John Wiley & Sons, New York, USA, 1991.
- [24] Cyert, R., March, J., Behavioral theory of the firm. In : Organizational behavior 2 – Essential theories of process and structure, Wiley-Blackwell, USA, 1992.
- [25] Die Welt, Internetquelle: www.welt.de/wirtschaft/webwelt/article13790867/So-haben-sich-Fernseher-in-80-Jahren-veraendert.html, Aufruf vom 25.2.2014.
- [26] Dietterich, T. G., Machine Learning, In: Nature Encyclopedia of Cognitive Science. Macmillan, London, 2003.
- [27] Döbel, C., Optimierung von Energieverbrauch und Wohnkomfort mit Methoden des maschinellen Lernens und der Entscheidungspsychologie, Energy Ecosystems Conference, Leipzig, 2014.

- [28] Double Edge Radio Productions, Internetquelle: www.doubleedge-radioproductions.org/radio.html, Aufruf vom 25.2.2014.
- [29] Drescher, G. L., Made-up Minds, a Constructivist Approach to Artificial Intelligence. MIT Press, Cambridge, USA, 1991.
- [30] Eisenführ, F., Weber, M., Langer, T., Rationales Entscheiden, Springer-Verlag Heidelberg Berlin, 2010.
- [31] Elman, J. L., Finding structure in time. In: Cognitive Science, Vol. 14, pp. 179-211, Austin, USA, 1990.
- [32] Eltako Electronics, GFVS-Safe II, Internetquelle: www.eltako.com/fileadmin/downloads/de/Datenblatt/FUNK_datenblatt_GFVS-SafeII.pdf, Aufruf vom 24.2.2014.
- [33] EPBD, Implementing the Energy Performance of Buildings Directive - Featuring Country Reports 2012, Porto, Portugal, 2013.
- [34] eQ-3/ ELV, Home Matic, Internetquelle: www.homematic.com/, Aufruf vom 24.2.2014.
- [35] Feldbaum, A. A., Dual control theory III-IV, In: Autonm Remote Control, 22, pp. 1-12, 109-121, Lund Institute of Technology, Schweden, 1961.
- [36] Freeman, J. A.; Skapura, D. M.: Neural Networks – Algorithms, Applications and Programming Techniques, Addison-Wesley, Bonn, 1991.
- [37] Friedrich, T. A., EU-Kommission will die Energieeffizienz künftig um ein Drittel erhöhen, VDI nachrichten vom 1.8.2014, VDI Verlag GmbH, Düsseldorf, 2014.
- [38] Gerdes, I., Klawonn, F., Kruse R., Evolutionäre Algorithmen, Vieweg Verlag, Wiesbaden, 2004.
- [39] Ghahramani, Z., Unsupervised Learning, University College London, UK, 2004.
- [40] Gira, Homeserver, Internetquelle: www.gira.de/homeserver, Aufruf vom 27.2.2014.
- [41] Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Boston, USA, 1953.
- [42] Grefenstette, J. J., Deception Considered Harmful, In: Foundations of Genetic Algorithms 2, Morgan Kaufmann, San Mateo, USA, 1993.
- [43] Gross, H.-M., Vorlesung – Lernen in kognitiven Systemen (SS2012), Technische Universität Ilmenau, 2012.

- [44] Hager, domovea, Internetquelle: www.hager-zuhause.de, Aufruf vom 27.2.2014.
- [45] Hamzeh, A., Rahmani, A., A Fuzzy System to Control Exploration Rate in XCS, In: Learning Classifier Systems, International Workshops, IWLCS 2003-2005, pp. 115-127, Springer-Verlag Berlin Heidelberg, 2007.
- [46] Hartbrich, I., Menschen haben es nicht so mit Nachkommastellen, vom 1.8.2014, VDI nachrichten, VDI Verlag GmbH, Düsseldorf, 2014.
- [47] Heine, B.-O., Hirsch, B., Hufschlag, K., Lesch, M., Meyer, M., Müller, R., Paefgen, A., Pieroth, G., Zur Modellierung ökonomischer Akteure mit begrenzten kognitiven Fähigkeiten – Anleitung zu einer problemspezifischen Ausdifferenzierung des Homo oeconomicus, WHU-Forschungspapier Nr. 110, Wissenschaftliche Hochschule für Unternehmensführung, Vallendar, 2006.
- [48] Hering, E., Triemel, J., Blank, H.-P., Qualitätsmanagement für Ingenieure, 4., überarbeitete Auflage, Springer-Verlag Heidelberg Berlin, 1999.
- [49] Herrmann, M., Natural Computing. Lecture 4 – The Schema Theorem and the Building Block Hypothesis. Informatics Forum 1.42, Edinburgh, UK, 2011.
- [50] Hoffmann, J., Vorhersage und Erkenntnis [Anticipation and Cognition], Göttingen, 1993.
- [51] Holland, J. H., Adaptation in Natural and Artificial Systems, In: University of Michigan Press, Ann Arbor, 2. edition, USA, 1994.
- [52] Holland, J. H., Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems, In: Machine Learning: An Artificial Intelligence Approach, vol. 2, Morgan Kaufman, Los Altos, CA, 1986.
- [53] Holland, J. H., Properties of the Bucket Brigade, In: Proceedings of the First International Conference on Genetic Algorithms and their Applications, Lawrence Erlbaum Associates Hillsdale, New Jersey, USA, 1985.
- [54] Holland, J. H., Reitman J. S., Cognitive Systems Based on Adaptive Algorithms, In: Pattern-Directed Inference Systems, Academic Press, New York, USA, 1978.
- [55] Holmes, J. H., Detection of Sentinel Predictor-Class Associations with XCS: A Sensitivity Analysis, In: Learning Classifier Systems, International Workshops, IWLCS 2003-2005, pp. 270-281, Springer Verlag Berlin Heidelberg, 2007

- [56] Holmes, J. H., Lanzi, P. L., Stolzmann, W., and Wilson, S. W., Learning classifier systems new models, successful applications, In: Information Processing Letters, vol. 82, no. 1, pp. 23-30, Elsevier Sciences, Amsterdam, 2002.
- [57] Hufschlag, K., Informationsversorgung lernender Akteure, Springer, eBook, 2009.
- [58] IBM, Autonomic Computing - Adoption Model, Internetquelle: http://www-03.ibm.com/autonomic/about_get_model.html, Aufruf am 24.02.2008
- [59] Institut für Gebäudetechnologie, Internetquelle: <http://www.gebaeude-iq.de/bierdeckel/Wohngebaeude/index.html>, Aufruf am 23.1.2014
- [60] James, W., The Principles of Psychology (Vol. 2), Harvard University Press, Cambridge, USA. 1981 (orig. 1890).
- [61] Jungermann, H., Die Psychologie der Entscheidung, Elsevier, Spektrum, Akad. Verlag, Heidelberg, 2005.
- [62] Kandel, E. R., Schwartz, J. H., Jessell, T. M., Neurowissenschaften. Eine Einführung, Spektrum Akademischer Verlag, Heidelberg, 1995.
- [63] Kharbat, F., Bull, L., Odeh M., Revisiting genetic selection in the XCS learning classifier system, In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 3, IEEE Press, Piscataway, 2005.
- [64] Kohonen, T., Self-organized formation of topologically correct feature maps, In: Biological Cybernetics, 43, 59-69, Springer-Verlag Berlin Heidelberg, 1982.
- [65] Krallmann, H., Schönherr, M., Trier, M., Systemanalyse in Unternehmen, Oldenbourg Wissenschaftsverlag, München, 2007.
- [66] Kratzer, K.-P.: Neuronale Netze. Grundlagen und Anwendungen, 2. Auflage, Hanser, München, 1990.
- [67] Wikimedia: <https://upload.wikimedia.org>, Aufruf am 13.7.2016.
- [68] Kruse, R., Moewes, C., Evolutionäre Algorithmen. Das Schematheorem. Otto-von-Guericke-Universität Magdeburg, 2011.
- [69] Kunz, J., Wissensmanagement und organisationales Lernen – Eine simulationsgestützte Analyse, Hamburg, 2006.
- [70] Landau, S., Sigaud, O., Picault, S., Gerard, P., An Experimental Comparison Between ATNoSFERES and ACS, In: IW LCS 2003-2005, LNAI 4399, pp. 144,160, Springer Verlag Heidelberg Berlin, 2007.

- [71] Lanzi, P. L., An Analysis of the Memory Mechanism of XCSM, In: Proceedings of the Third Genetic Programming Conference, Morgan Kaufmann, San Mateo, USA, 1998.
- [72] Lanzi, P. L., Learning Classifier Systems from a Reinforcement Learning Perspective, Springer-Verlag Heidelberg Berlin, 2002.
- [73] Lee, E. A., Cyber-Physical Systems – Are Computing Foundations Adequate?, In: NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Austin (Texas), 2006.
- [74] Li, W.-H., Molecular Evolution, In: Sinauer Associates, Sunderland, MA, USA, 1997.
- [75] Liepins, G. E., Hilliard, M. R., Palmer, M., Rangarajan, G., Alternatives for Classifier System Credit Assignment, In: Proceedings of the 11th international joint conference on Artificial Intelligence, Morgan Kaufman Publishers, San Francisco, CA, USA, 1989.
- [76] Lin, L. J., Mitchell, T. M., Memory approaches in reinforcement learning in non-markovian domains. Technical Report CMU-CS-92-138, Carnegie Mellon University, School of Computer Science, 1992.
- [77] Lindenberg, S. Die Methode der abnehmenden Abstraktion, In: Modellierung sozialer Prozesse, Informationszentrum Sozialwissenschaften, Bonn, 1991.
- [78] Louis, S. J., Rawlins, G. J. E., Syntactic Analysis of Convergence in Genetic Algorithms, In: Foundations of Genetic Algorithms, 141-151, San Mateo, CA, USA, 1993.
- [79] Loxone, Miniserver, Internetquelle:
www.loxone.com/dede/miniserver/features.html, Aufruf vom 27.2.2014.
- [80] Macho, S., Kognitive Modellierung mit Neuronalen Netzen. Eine anwendungsorientierte Einführung, Huber, Bern, 2002.
- [81] Marín-Blázquez, J. G., Schulenburg, S., A Hyper-Heuristic Framework with XCS: Learning to Create Novel Problem-Solving Algorithms Constructed from Simpler Algorithmic Ingredients, In: Learning Classifier Systems, International Workshops, IWLCS 2003-2005, pp. 193-218, Springer Verlag Berlin Heidelberg, 2007.
- [82] McCallum, R. A., Reinforcement Learning with Selective Perception and Hidden State, PhD thesis, University of Rochester, Rochester, USA, 1995.

- [83] McCulloch, W., Pitts, W., A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, Kluwer Academic Publishers, Alphen aan den Rijn, NL, 1943.
- [84] McLeod, P., Plunkett, K., Rolls, E. T., Introduction to connectionist modelling of cognitive processes. University Press, Oxford, UK, 1998.
- [85] Meuleau, N., Peshkin, L., Kim, K. E., Kaelbling, L. P., Learning finite-state controllers for partially observable environments. In: Fifteenth Conference on Uncertainty in Artificial Intelligence. AAAI, Morgan Kaufmann, San Francisco, USA, 1999.
- [86] Minsky, M., Steps toward artificial intelligence, In: Proceedings of the IRE, vol. 49, no. 1, McGraw-Hill, 1963.
- [87] Mitchell, T. M., An Introduction to Genetic Algorithms, Massachusetts Institute of Technology, MIT Press, USA, 1996.
- [88] Mitchell, T. M., Carbonell, J. G., Michalski, R. S., Machine Learning: A Guide to Current Research, Springer, 1997.
- [89] Morik, K., Maschinelles Lernen, In: Einführung in die Künstliche Intelligenz, S. 243-297, Addison-Wesley, Bonn, 1995.
- [90] Mostaghim, S., Schmeck, H., Organic Computing - Self-Organisation and Emergence, Internetquelle: <http://www.aifb.uni-karlsruhe.de/Lehre/Sommer2007/OC/unterlagen/OC2.a.pdf>, Aufruf am 24.02.2008
- [91] Mozer, M. C., Lessons from an Adaptive Home. In: D. Cook & S. Das (Eds.), Smart environments: Technologies, protocols, and applications, J. Wiley and Sons, Colorado, 2004.
- [92] Muhl, G., Werner, M., Jaeger, M. A., Herrmann, K., Parzyjega, H., On the Denition of Self-Managing and Self-Organizing Systems, Internetquelle: http://www.kbs.cs.tu-berlin.de/publications/fulltext/kivs2007_2.pdf, Aufruf am 24.02.2008.
- [93] MyGekko, Internetquelle: www.my-gekko.com, Aufruf vom 27.2.2014.
- [94] Nicolai, A. Der „trade off“ zwischen „rigour“ und „relevance“ und seine Konsequenzen für die Managementwissenschaften, In: Zeitschrift für Betriebswirtschaft (ZfB), 74. Jg. Nr. H. 2, Wiesbaden, 2004.
- [95] Norm, DIN EN 15232:2012, Energieeffizienz von Gebäuden – Einfluss von Gebäudeautomation und Gebäudemanagement, 2012.

- [96] Nuissl, N., Geräte im Haus sollen bald über neue Plattform kommunizieren, In: VDI Nachrichten, VDI Verlag GmbH, Köln, 2013.
- [97] Oberhofer, W., Zimmerer, T., Wie Künstliche Neuronale Netze lernen: Ein Blick in die Black Box der Backpropagation Netzwerke, In: Regensburger Diskussionsbeiträge Nr. 287, Regensburg, 1996.
- [98] Papadakis, A., Internetquelle: <http://www.psychology48.com/deu/d/nutzen/nutzen.htm>, Aufruf vom 10.5.2014.
- [99] Pfeifer, T., Qualitätsmanagement – Strategien, Methoden, Techniken, 3., völlig überarbeitete Auflage, Carl Hanser Verlag München Wien, 2001.
- [100] Popper, K. R., Lesebuch: ausgewählte Texte zur Erkenntnistheorie, Verlag C. H. Beck, Tübingen, 2000.
- [101] Pospeschill, M., Konnektionismus und Kognition. Eine Einführung, Kohlhammer, Stuttgart, 2004.
- [102] Rademacher, Homepilot, Internetquelle: homepilot.rademacher.de/, Aufruf vom 27.2.2014.
- [103] Rey, G. D., Wender, K. F., Neuronale Netze. Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung, Verlag Hans Huber, Bern, 2008.
- [104] Rojas, R., Neural Networks, Springer-Verlag, Berlin, 1996.
- [105] Rumelhart, D. E., Hinton, G. E., McClelland, J. L., A General Framework for Parallel Distributed Processing, In: Parallel Distributed Processing, MIT Press, San Diego, USA, 1986.
- [106] RWE Effizienz GmbH, Smarthome, Internetquelle: www.rwe-smarthome.de/web/cms/de/448330/smarthome, Aufruf vom 24.2.2014.
- [107] Sander, S., Krödel, M., Krause, H., Diplomarbeit: Ermittlung des Energieeinsparpotenzials durch Gebäudeautomation in Wohngebäuden anhand verschiedener Wohnsituationen, Hochschule Rosenheim, 2012.
- [108] Schäffer, U., Kontrolle als Lernprozess, Gabler Edition Wissenschaft, Wiesbaden, 2001.
- [109] Schmidt, G., Bitkom, IHS Technology, 2014.
- [110] Schöneburg, E., Hansen, N., Gawelczyk, A., Neuronale Netze: Einführung, Überblick und Anwendungsmöglichkeiten, Markt & Technik Verlag, Burghann, 1990.

- [111] Siemens, Gebäudeautomation – Einfluss auf die Energieeffizienz. Anwendung gemäß EN 15232:2012 eu.bac Produktzertifizierung, Deutschland, 2012.
- [112] Simon, A. H., Models of Man. Presents mathematical models of human behaviour, John Wiley and Sons, New York, USA, 1957.
- [113] Somfy, TaHoma/ io-homecontrol, Internetquelle: www.somfy.de/home/produkte/produkte/io-homecontrol-tahoma.html<http://www.loxone.com/dede/miniserver/features.html>, Aufruf vom 27.2.2014.
- [114] Statistisches Bundesamt, Internetquelle: www.destatis.de, „“, Aufruf am 1.1.2015.
- [115] Stein, B., Modellierung wissensintensiver Probleme, Vorlesung an der Universität Weimar, Weimar, 2014.
- [116] Stempfle, C., Einführung in Organic Computing, In: Konzepte und Techniken in Organic Computing, Universität Augsburg, 2008.
- [117] Stolzmann, W., Anticipatory Classifier Systems, In: Proceedings of the Third Annual Conference on Genetic Programming, University of Wisconsin, Morgan Kaufmann, Wisconsin, USA, 1998.
- [118] Stolzmann, W., Antizipative Classifier Systeme [Anticipatory Classifier Systems], Shaker Verlag, Aachen, Osnabrück, 1997.
- [119] Stolzmann, W., Latent Learning in Khapera Robots with Anticipatory Classifier Systems, In: Proceedings in the 1999 Genetic and Evolutionary Computation Conference (GECCO'99), Orlando, USA, 1999.
- [120] Strecker, S., Künstliche Neuronale Netze – Aufbau und Funktionsweise, Arbeitspapiere, Justus-Liebig-Universität, Gießen, 1997.
- [121] Suchanek, A., Ökonomischer Ansatz und theoretische Integration, Bd. 84 der Reihe Die Einheit der Gesellschaftswissenschaften, Tübingen, 1994.
- [122] Sutton, R. S., DYNA, an integrated architecture for learning, planning, and reacting. In: Working Notes of the 1991 AAAI Spring Symposium on Integrated Intelligent Architectures, pp. 151-155, Morgan Kaufmann, San Francisco, USA, 1991.
- [123] Telefunken, Joonior, Internetquelle: www.telefunken-sb.de/, Aufruf am 24.2.2014.
- [124] Thagard, P.. Mind: Introduction to Cognitive Science. 2nd ed. Cambridge, MIT Press, USA, 2005.

- [125] Tolman, E. C., Purposive Behavior in Animals and Men. Appleton, New York, USA, 1932.
- [126] Tomlinson, A., Bull, L., A zeroth level corporate classifier system. In: Learning Classifier Systems. From Foundations in Applications. Volume 1813 of Lecture Notes in Artificial Intelligence, 306-313, Springer-Verlag Heidelberg Berlin, 2000.
- [127] Tomlinson, A., Bull, L., CXCS In: Learning Classifier Systems: from Foundations to Applications, 194-208, Springer-Verlag Heidelberg Berlin, 2000.
- [128] Vainio A.-M., Learning and adaptive fuzzy control system for smart home. Developing Ambient Intelligence. Tampere, Finland: Springer-Verlag France, Paris, Frankreich, 2006.
- [129] Vanberg, V. J., The rationality postulate in economics: its ambiguity, its deficiency and its evolutionary alternative, In: Journal of Economic Methodology, 11. Jg., Nr. 1., S. 1-29., Taylor and Francis Journals, UK, 2004.
- [130] Vöcking, B., Alt, H., Dietzfelbinger, M., Reischuk, R., Scheideler, C., Vollmer, H., Wagner, D., Taschenbuch der Algorithmen, Springer-Verlag Berlin Heidelberg, 2008.
- [131] Wada, A., Takadama, K., Katsunori, S., Osamu, K., Analyzing Parameter Sensitivity and Classifier Representations for Real-Valued XCS, In: Learning Classifier Systems, International Workshops, IWLCS 2003-2005, pp. 1-16, Springer-Verlag Berlin Heidelberg, 2007.
- [132] Watkins, C. J. C. H., Dayan, P., Q-learning. Machine Learning 8(3): 272-292, Kluwer Academic Publishers, Alphen aan den Rijn, NL, 1992.
- [133] Wernstedt, J., Experimentelle Prozessanalyse, Oldenbourg Wissenschaftsverlag, München, 1989.
- [134] Widrow, B., Hoff, M. E., Adaptive switching circuits, In: IRE WESCON Convention Report, vol. 4, pp. 709-717, Stanford, 1960.
- [135] Wiering, M., Schmidhuber, J., HQ-Learning. Adaptive Behavior 6, Computer Based Learning Unit, University of Leeds, UK, 1997.
- [136] Wikipedia, Self-management (computer science), Internetquelle: http://en.wikipedia.org/wiki/Self-management_%28computer_science%29, Aufruf am 8.2.2014.

- [137] Wilson, S. W., Classifier fitness based on accuracy, In: *Voluntary Computation*, vol. 3, no. 2, pp. 149-175, MIT Press Cambridge, MA, USA, 1995.
- [138] Wilson, S. W., Classifiers that Approximate Functions, In: *Natural Computing*, Edinburgh, UK, 2001.
- [139] Wilson, S. W., Get real! XCS with continuous-valued inputs, In: *Learning Classifier Systems: from Foundations to Applications*, lecture Notes in Artificial Intelligence (LNAI-1813), Springer-Verlag Berlin Heidelberg, 2000.
- [140] Wilson, S. W., Goldberg, D. E., A Critical Review of Classifier Systems, In: *Proceedings of the Third International Conference on Genetic Algorithms*, S. 244-255, George Mason University, San Francisco, USA, 1989.
- [141] Wilson, S. W., Knowledge growth in an artificial animal, In: *Proceedings of the 1st International Conference on Genetic Algorithms and Their Application*, pp. 16-23, Springer-Verlag Berlin Heidelberg, 1985.
- [142] Wilson, S. W., Sigaud, O., Learning classifier systems: a survey, In: *Soft Computing*, 11:1065-1078, Springer-Verlag Heidelberg Berlin, 2007.
- [143] Wineberg, M., Oppacher, F., Distance between Populations, In: *Genetic and Evolutionary Computation – GECCO 2003*, pp. 1481 – 1492, Springer-Verlag Berlin Heidelberg, 2003.
- [144] Wineberg, M., Oppacher, F., The Underlying Similarity of Diversity Measures Used in Evolutionary Computation, In: *Genetic and Evolutionary Computation – GECCO 2003*, pp. 1493 – 1504, Springer-Verlag Berlin Heidelberg, 2003.
- [145] Wittenmark, B., Adaptive Dual Control Methods: An overview, Lund Institute of Technology, Schweden, 2002.
- [146] Zeise, N., Lehrbrief: Prozessmodellierung und Prozessmanagement, Schmalkalden, 2012.
- [147] Zell, A.: *Simulation Neuronaler Netze*, 1. unveränderter Nachdruck, Addison-Wesley, Bonn, 1996.

